# Investigating the Storage Capacity of a Network with Cell Assemblies

Richard Bowles

Submitted to the Middlesex University in partial fulfilment of requirements for the degree of Doctor of Philosophy

School of Computing Science,
Middlesex University,
The Burroughs, London
NW4 4BT
United Kingdom

April 20, 2006

# Contents

# List of Figures

# List of Tables

## Abstract

Cell assemblies are co-operating groups of neurons believed to exist in the brain. Their existence was proposed by the neuropsychologist D.O. Hebb who also formulated a mechanism by which they could form, now known as Hebbian learning. Evidence for the existence of Hebbian learning and cell assemblies in the brain is accumulating as investigation tools improve. Researchers have also simulated cell assemblies as neural networks in computers.

This thesis describes simulations of networks of cell assemblies. The feasibility of simulated cell assemblies that possess all the predicted properties of biological cell assemblies is established. Cell assemblies can be coupled together with weighted connections to form hierarchies in which a group of basic assemblies, termed *primitives* are connected in such a way that they form a *compound* cell assembly. The component assemblies of these hierarchies can be *ignited* independently, *i.e.* they are activated due to signals being passed entirely within the network, but if a sufficient number of them are activated, they co-operate to ignite the remaining primitives in the compound assembly.

Various experiments are described in which networks of simulated cell assemblies are subject to external *activation* involving cells in those assemblies being stimulated artificially to a high level. These cells then *fire*, *i.e.* produce a spike of activity analogous to the spiking of biological neurons, and in this way pass their activity to other cells. Connections are established, by learning in some experiments and set artificially in others, between cells within primitives and in different ones, and these connections allow activity to pass from one primitive to another. In this way, activating one or more primitives may cause others to ignite. Experiments are described in which spontaneous activation of cells aids recruitment of uncommitted cells to a neighbouring assembly. The strong relationship between cell assemblies and Hopfield nets is described.

A network of simulated cells can support different numbers of assemblies depending on the complexity of those assemblies. Assemblies are classified in terms of how

many primitives are present in each compound assembly and the minimum number needed to complete it. A 2-3 assembly contains 3 primitives, any 2 of which will complete it. A network of $N$ cells can hold on the order of $N$ 2-3 assemblies, and an architecture is proposed that contains $O(N^2)$ 3-4 assemblies. Experiments are described that show the number of connections emanating from each cell must be scaled up linearly as the number of primitives in any network increases in order to maintain the same mean number of connections between each primitive. Restricting each cell to a maximum number of connections leads to severe loss of performance as the size of the network increases. It is shown that the architecture can be duplicated with Hopfield nets, but that there are severe restrictions on the carrying capacity of either a hierarchy of cell assemblies or a Hopfield net storing 3-4 patterns, and that the promise of $N^2$ patterns is largely illusory. When the number of connections from each cell is fixed as the number of primitives is increased, only $O(N)$ cell assemblies can be stored.

# Acknowledgement

I would like to thank my director of studies, Dr. Christian R. Huyck, for all his invaluable help and his patience in guiding me. I would also like to thank my supervisor, Dr. Siri Bavan, for suggesting useful directions of research. Without both these men, this thesis would not have been possible.

# Chapter 1

# Introduction

The mammalian brain consists of a vast number of nerve cells, or *neurons*, with estimates varying from about $10^9$ [155] to about $10^{11}$ [36, 187]. Cell assemblies are networks of cells that are believed to exist within the mammalian brain each incorporating many cells to form closed circuits of cells with certain specific properties. The supposed presence of cell assemblies allows many more pieces of information to be stored than there are individual cells in the brain [153] and permits more complex structures such as hierarchies of concepts to develop. The fact that cell assemblies provide an elegant mechanism by which these structures can form is one of several indirect arguments for their existence, although empirical evidence for their existence is starting to appear.

Connectionist researchers are investigating the properties of simulated cell assemblies in computer programs, and several such simulations are the main areas of interest in this thesis. However, unlike standard connectionist architectures such as Multi-Layer Perceptrons (MLPs) and Kohonen nets, there are few generally accepted principles. Each researcher uses a slightly different set of equations and constructs his or her architecture in a slightly different way. This may be because of the difficulty of specifying brain function in mathematical form. Certainly different mathematical models of neurons exist [31, 96], all of which are only approximations to the exact behaviour of neurons. The more complex such a model is, the more processing it requires and the less it lends itself to simulation. Often in order to simulate networks of hundreds or thousands of cells in reasonable time the cell model must be stripped

to its bare essentials. It is only really necessary to use a complex model of a neuron when modelling brain function. After all, neural networks are used either for information processing or to model the brain, and standard neural network architectures have shown that complicated models of neurons are unnecessary to provide useful results in information processing applications. This is certainly the case with MLPs or Kohonen nets - they were inspired by the brain, but designed to produce useful results in AI (Artificial Intelligence) that were difficult to obtain any other way - and should be the case with simulated cell assemblies also. Unfortunately, while the list of practical applications of standard neural architectures is large (e.g. NETTALK [157] that reads English text out loud, or WISARD [183] that scans CCTV images for intruders), I can recall only two established practical application for simulated cell assemblies: Knoblauch *et al.* [103] have applied simulated cell assemblies to decoding sequences of spoken commands to a simulated robot, and Huyck and Orengo [94] have used cell assemblies to perform categorisation of US senators into Democrat and Republican based on voting records, and information retrieval to match documents in a database.

Cell assemblies illustrate two dynamics, one short-term, the other long-term, and thus offer possible explanations of both short-term memory and long-term memory. The short-term dynamic is ignition, when the number of neurons that fire in the cell assembly becomes sufficient to allow persistent firing of the cell assembly even when those neurons have fatigued, *i.e.* the level of firing persists even when individual neurons cease to fire. The long-term dynamic is learning, in which weighted connections between neurons strengthen or weaken so that cell assemblies more readily respond to stimulation in future in a meaningful way. Learning allows more complex phenomena to evolve, such as one cell assembly igniting another, and larger neural structures such as hierarchies of cell assemblies [83]. In this thesis, simple, indivisible cell assemblies are referred to as *primitives* and the hierarchies of primitives are referred to as *compound cell assemblies*. Individual cells may belong to no more than one primitive, although they can belong to more than one compound assembly by virtue of the fact that compound assemblies can overlap.

Researchers have posited that cell assemblies in the brain are responsible for a

range of cognitive processes, including simple association and sequential processing [57, 29]. Thus, cell assemblies offer an intermediate step between the simplest level of cognition, which is neural activity, and the higher level of symbol processing. Certainly experiments show that simulated cell assemblies can display a variety of useful actions, from categorising a broad range of patterns [90] to variable binding [168, 46].

This thesis gives a basic introduction to cell assemblies and shows that they can evolve naturally from a simple system of simulated cells to which a few rules are applied. It also demonstrates that the storage capacity of a network of cell assemblies, in terms of the number of compound cell assemblies that can be constructed from $N$ primitives increases in proportion to $N^2$, and that, in principle, networks can be constructed with even higher capacities. The closest "relative" that cell assemblies have in the family of standard neural network architectures is the Hopfield net. There are essential differences between the two architectures, but they have more things in common than separate them. For this reason, a comparison between cell assemblies and Hopfield nets is described. Experiments indicate that, in spite of various researchers who state that the storage capacity of Hopfield nets, in terms of the number of patterns that can be stored and reliably retrieved, is linearly proportional to the number of cells ($n$) within them, Hopfield nets can be constructed that can exist in stable states of the order of $n^2$, $n^3$ or even higher powers. However, there is a distinction between the number of stable states in a network and the number of patterns that can usefully be stored, and this distinction is made clear.

## 1.1 Hebbian Learning and the Concept of Cell Assemblies

The concept of cell assemblies owes its origins to Hebb in his book Organization of Behavior [74], although the suggestion that information may be stored in the brain in the strength of connections between neurons had already been made by Tanzi [182] and Ramon y Cajal [147]. Wernicke had also proposed various ideas that might be considered the forerunners to the theory of cell assemblies [57]. However, the credit

for the fully developed idea really belongs to Hebb. The principle that Hebb proposed, now known as Hebbian learning, is, to date, the only learning mechanism that has been experimentally verified to occur in the nervous system [72]. It states that connections between neurons in the brain strengthen as a result of coincidental firing. Hebb further implied that neurons would organise themselves into cell assemblies as a natural result of this.

In the 1940s the techniques available to neuroscientists were limited [197]. They could study the effects that brain damage had on behaviour, whether caused by physical injuries to the brain, by strokes or by other causes such as Alzheimer's disease. However, such an analysis was only possible if the brain of the victim became available after death. This was because non-invasive techniques, such as Positron Emission Tomography (PET) and Magnetic Resonance Imaging (MRI) had not yet been developed.

Bearing in mind the absence of investigative techniques, it is perhaps surprising that the neuropsychologist Donald Hebb should produce a theory which attempted to explain some aspects of human behaviour in terms of neurons and the connections between them. It is also surprising that this theory should have survived to the present day, when evidence for or against the theory can be obtained much more easily.

### 1.1.1 Hebbian learning

In Organization of Behavior, Hebb proposed the idea of Hebbian learning. This predicted that the connection between two neurons would change in strength, depending on the activity of the neurons themselves. Such a connection is called a *synapse* and it is responsible for the transmission of sodium and potassium ions from the axon of one neuron to the dendrites of another. In brief, it stated that if both the neuron providing the signal transmitted across the synapse (the pre-synaptic neuron) and the neuron accepting the signal (the post-synaptic neuron) happened to fire simultaneously, or at least within a very short space of time, then the synaptic connection strength would increase. This would increase the future likelihood of the post-synaptic neuron firing whenever the pre-synaptic neuron fired. An analogy for

this is the strengthening of a muscle through repeated use. Since Hebb formulated this idea, its biological correlate in the brain, Long Term Potentiation (LTP), has been established [14, 100]. Larson and Lynch [112] have shown that Hebbian learning occurs for any synapse in which the post-synaptic neuron fires within 50 ms of the pre-synaptic neuron, and Dayan and Abbott [41] have shown that Hebbian learning takes place in the hippocampus, cerebellum and visual system. The name LTP has been attached to the strengthening of connections in cell assembly simulations, and the adaptation of simulated synaptic connections seen on a longer time scale is also referred to as Hebbian learning.

Let us assume then that the persistence or repetition of a reverberatory activity (or "trace") tends to induce lasting cellular changes that add to its stability. The assumption can be precisely stated as follows: When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased [74]. Hebb used the principle of hearing a clock strike twelve to illustrate this. The fact that we hear the clock strike twelve, rather than strike one twelve times, implies that some change must have been wrought in the connections in the brain between one strike of the clock and the next. Of course, it does not necessarily follow that the change in the brain must be Hebbian in nature.

Hebb's theory has subsequently been adapted to include the weakening of synaptic connections. There is biological evidence that indicates that the synaptic strengths may decrease whenever only one of the neurons is inactive [178, 194]. Such an effect is referred to as Long Term Depression (LTD). Although the exact nature of LTD in the brain is not known, when applied to simulated cell assemblies, there are two possible types: *post-not-pre LTD* and *pre-not-post LTD*. Post-not-pre LTD, used by Hetherington and Shapiro [78] for example, occurs when the post-synaptic neuron is active but the pre-synaptic neuron is not. Hebb's term, *heterosynaptic* LTD, is generally taken to refer to pre-not-post LTD, in which the pre-synaptic neuron is active but the post-synaptic neuron is not. The effect of these two different types of LTD is summarised in figure 1.1 on page 7.

Post-not-pre LTD.

| | Pre-synaptic cell active | Pre-synaptic cell inactive |
|---|---|---|
| Post-synaptic cell active | connection strengthens | connection weakens |
| Post-synaptic cell inactive | no change in strength | no change in strength |

Pre-not-post LTD.

| | Pre-synaptic cell active | Pre-synaptic cell inactive |
|---|---|---|
| Post-synaptic cell active | connection strengthens | no change in strength |
| Post-synaptic cell inactive | connection weakens | no change in strength |

Figure 1.1: Both types of LTD involve weakening of connection strengths, although under slightly different circumstances.

### 1.1.2 Cells assemblies result from Hebbian learning

Hebb extended his theory as follows. He proposed that groups of neurons would form reverberating circuits that would be stimulated by external inputs, and would then continue to reverberate, even after the stimulus had been removed. As neurons forming part of this circuit ceased to be active, they would be reactivated by other neurons in the circuit. The analogy that Hebb used was that such a circuit would act like a ringing bell, which continues to sound even after the hammer that caused it to ring had been removed. These reverberating circuits Hebb termed cell assemblies. Some researchers believe that cell assemblies follow as the inevitable consequence of Hebbian learning [15]. Singer [166] suggests that the Hebbian learning is responsible for cell assemblies forming between different cortical columns in the visual cortex that encode frequently occurring groups of features for ease of detection.

Figure 1.2 has been duplicated from Hebb's 1949 book. Each of the arrows represents a group of neurons. The numbers indicate the order in which they become active, with some groups of neurons firing more than once in the complete cycle.

The most important simulation of cell assemblies in a computer in the few years after Hebb formulated his theory was the work of Rochester *et al.* [150] in which a network of simulated cells was used in an attempt to set up a cell assembly. Unfortunately, this simulation had mixed results. Although simulated neurons did show synchronous firing, *i.e.* the assemblies that formed showed the effect of *dynamic*

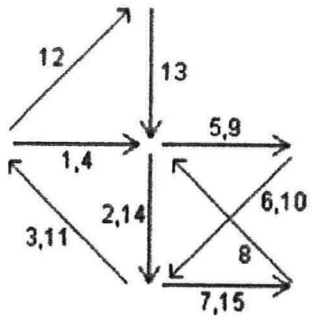


Figure 1.2: Hebb's concept of a Cell Assembly (reproduced from [74])

*completion* as described below, their activity did not persist beyond the removal of the external stimulus. MacGregor and McMullen [119] replaced the simple cell model with one modelling spiking motor neurons more realistically. Like Rochester *et al.*, they achieved concurrent spiking but not the sustained activity lasting approximately 500ms that Hebb had hypothesized [74], p. 74. Lansner and Fransen in turn replaced the motor neuron model with one for pyramidal cells [111] which allowed activity to be sustained after external stimulation was removed.

Other notable simulations of cell assemblies include Hetherington and Shapiro [78] and Iglesias *et al.* [95], both of which show that cell assemblies can form spontaneously as a result of stimulating a network of randomly connected cells which obey simple rules of Hebbian learning. Hetherington and Shapiro report simple pattern recognition, in which four non-overlapping patterns of bits were successfully recognised by the creation of four cell assemblies in the same network. These assemblies could be activated independently, even by the application of noisy test patterns. They claim that for the assemblies to form, it is necessary to use post-not-pre LTD and *dendritic partitioning* of inputs. Their neural networks used a series of twenty inputs connected to certain cells within the 360 cell network. The presence of LTD could easily lead to the weakening of these connections, as the cell assemblies could be active in the absence of external input, and learning was always present in the network. For this reason, the learning rule could not be used to adapt connection strength from the inputs to the network, and this led to the input connections being treated separately (the dendritic partitioning referred to) [177].

One of the researchers that influenced Hebb was Lashley [113]. His idea - to be fair, based on rather little evidence - was that memory was based on waves of activation that spread across the brain. However, even such an idea does not rule out the possibility of cell assemblies. Beurle [12] showed mathematically that a network of idealised brain cells could sustain waves of activity providing that they possessed the crucial property of fatigue, and gave some conditions under which such waves would propagate, die out, or reverse direction. His analogy was based upon cells firing, passing activity to their neighbours, as a result of which they fatigue and are unable to fire again immediately. The neighbouring cells pass activity to all their neighbours, but the fatigued cells do not fire, so the wave of cell firing only passes in one direction. This is illustrated in figure 1.3. By specifying areas of the network in which cells are fatigued prior to the wave passing, or areas in which cells have been primed so that they are almost ready to fire, Beurle showed that the network could demonstrate a variety of wave-like behaviour, such as reflection or refraction. Beurle's work stands out as the major paper to attempt to reconcile Lashley's wave activity with what was known at the time about cell assemblies. Lashley's ideas influenced the development of Pribram's Holonomic Brain theory, in which cognitive function is believed to derive from a matrix of wave interference patterns, in much the same way that a holographic image arises from the interference of light waves [143].

Hebb's theories have also been the influence for many more recent theories of cognition, including Abeles [1], Amit [6], Braitenberg [23], Damasio [37], Edelman [48], Marr [122], McGregor [129], Mesulam [130], Miller [132], Milner [134], Palm [138], Pulvermüller [145], Shaw et al. [164] and Wickelgren [186]. Cell assemblies have also appeared in simulations of language processing [102].

Variations on Hebbian learning include Signal Hebbian Learning and Differential Hebbian Learning [107]. Hebb's ideas have had a wide-ranging effect on neural networks. The equation for Hebbian learning exists in several similar forms (one of which is Equation 3.3 on page 59) and has formed the basis for updating weighted connections in various architectures, for instance, back-propagation for Multi-Layer Perceptrons, and Kohonen Self-Organising Topographical Maps [105].

It is difficult to obtain direct evidence of the existence of cell assemblies in the

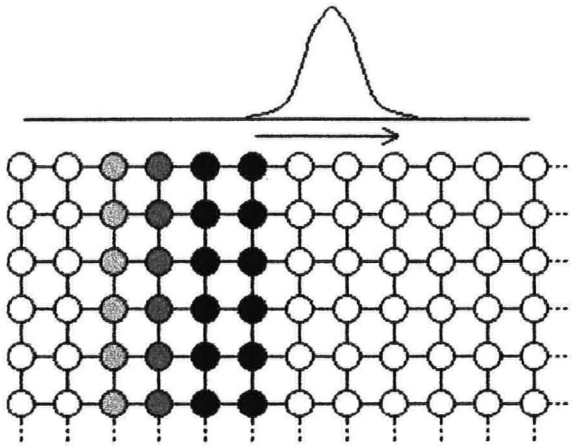

Figure 1.3: Beurle's wave analogy of cell activity. Cells fatigue when they fire (darkened circles) allowing activity to pass only one way. A wave of activity shown by the curve at the top passes in the direction of the arrow. Cells gradually recover ready for another wave of activity.

brain, although with recent techniques [43, 137, 148] the biological case for them is becoming stronger [169]. Instead, if cell assemblies are present in the brain, their existence must be inferred from indirect evidence. This is unfortunate, as it means that with our current techniques, their existence cannot be conclusively proven, and that other researchers can always argue that some other mechanism is responsible for the behavioural phenomena attributed to them. However, this does not stop some researchers (such as [142]) writing as though their existence were beyond doubt.

More common is indirect evidence from behavioural experiments, such as [146, 176]. The larger cell assemblies in the brain may well consist of many hundreds or thousands of neurons [146]. Each of these neurons has upwards of 2000 synapses and dendrites, each of which take the form of tiny filaments, many times thinner than a human hair. It would be a task requiring unimaginable precision and dexterity to untangle them. This Herculean task pales into insignificance compared to tracking down all the connections of neurons in the brain that may or may not form a cell assembly. It is generally accepted that forward connections from one area of the brain to another are generally matched by recurrent connections back to the area of origin [36], which raises the possibility that cell assemblies may be universal in the brain.

The evidence that does exist for cell assemblies takes the form of experiments which are carried out, for which cell assemblies are proposed as the best explanation for the results. Alternative explanations, such as recurrent networks which achieve attractor states, are generally not presented. A common approach is to look for *syn-firing*, the phenomenon of cells invariably firing in close synchronisation [123]. This does seem to imply that the cells have some connection, although it is not proof that they form part of a cell assembly. The technique used has improved to the point where the activity from several neurons can be monitored simultaneously in conscious animals [60, 144].

Sakurai [154] lists five properties that cell assemblies must possess in order to function in the way that Hebb describes:

1. Dynamic Completion. Activating a sufficient number of the cells in a cell assembly is sufficient to activate the majority of the cells.

2. Dynamic Persistence. Cells in each cell assembly remained active for a substantial number of time steps after external stimulation is removed, although it is not necessary for each cell to remain active for many time steps. Cells that fire become tired and stop firing, only to be re-ignited at later time steps by other cells in the same cell assembly. This allows cell assembly activity to persist beyond the point at which single cells fatigue.

3. Sparse Coding. Each cell assembly contains a minority of the cells in the entire network, although all the cells in the network may be committed to one or more cell assemblies.

4. Dynamic Construction. This is the property by which a network of cells is capable of learning cell assembly patterns by adapting the connection strengths between cells.

5. Overlapping set coding. The property that cells contribute to more than one cell assembly. Sakurai reasoned that the number of concepts that could be stored in the brain vastly outnumbers the number of brain cells present, so the

storing of concepts in the form of cell assemblies necessitates the sharing of cells between them.

He concludes that neurons must co-operate in order to represent knowledge and memories. He gives several reasons why this must be the case:

1. There is a virtually unlimited number of information items that need to be stored in the human memory. Not only are there single items, such as "dog" or "cat", but these can be joined to form an unlimited number of combinations (e.g. "Max is a large, hairy dog"). Even if every neuron in the human brain were dedicated to storing knowledge, there would not be enough neurons to encode all the possible information that humans have to deal with. Yet, humans do manage to store all these information items.

2. It is inefficient to store the similarities between items using totally separate neuronal codings. For example, Dylan the golden retriever and Merlin the golden retriever share many items in common, which must be represented in the brain of their owner in the form of neuronal codings. It would be highly inefficient to represent all the information known about the two dogs using totally separate codings, which implies that the two codings share a lot of the neurons in common (indeed in common with all the other dogs that the owner knows).

Instead, Sakurai suggests that information must be stored in groups of neurons using *population ensemble coding* [153], the sharing of neurons among several cell assemblies. He outlines three experiments performed on laboratory rats, which indicate that individual neurons must be present in more than one population ensemble. However, these experiments stop short of demonstrating that these groups of neurons must form reverberating circuits.

Palm [138] defines cell assemblies similarly, although he concentrates mainly on activation dynamics rather than structure. A group of neurons, $Y$, forms a cell assembly if activating a subgroup, $X$, causes the majority of cells in $Y$ to become active. Palm coined the term *ignition*, *i.e.* $X$ ignites $Y$. A group is said to be

*persistent* if the group maintains an activity level above a specified threshold after external stimulation has been removed. If a group of cells ignites without additional cells outside that group firing, it is said to be *invariant*. Group $Z$ *supports* group $Y$, if $Y$ is not persistent on its own but does persist when $Z$ ignites. This does not preclude $Y$ igniting in the presence of another group (e.g. $W$ and $Z$ may both support $Y$ if $Y$ can ignite in the presence of either $W$ or $Z$). A group of cells can be invariant while containing subgroups that support or ignite it, in which case it is a *Palm-assembly*. I adopt Palm's use of the word *ignition*, although the other terminology is less relevant in the context of this thesis. For this reason, Sakurai's definition is taken to be the point of reference in the following chapters rather than Palm's.

Scientists generally estimate the storage capacity of the human brain in terms of the number of bits of information, since the number of neurons it contains and the number of connections per neuron can only be estimated. If we assume that the brain contains somewhere between $10^9$ and $10^{11}$ neurons, and that each cell has about 20,000 synaptic connections (equivalent to 10,000 synapses for each neuron), then the brain contains somewhere between $10^{13}$ and $10^{15}$ synapses. Taking each synapse to represent a single binary value gives the same range of numbers for the number of bits that can be stored in the brain. However, each synapse has a variable firing threshold, which changes as the neuron is repeatedly activated. Assuming that each threshold can take 100 distinguishable values gives the storage capacity of the brain as between $10^{15}$ and $10^{17}$ bits. This figure does not represent the total number of *items* that can be remembered in the brain, since, unlike a digital computer, the brain does not remember items in terms of single binary digits. Regardless of the exact form in which memories are stored, each must consist of many bits of data. In 1666, Robert Hooke estimated his own memory storage by multiplying the speed at which he thought by his estimated lifespan, and producing a figure of $2 \times 10^9$ storage bits. A more realistic estimate was derived by Landauer [109] from a range of psychological experiments. He estimated the storage capacity to be approximately one billion distinct memories. He later revised this esimate to $4 \times 10^8$.

Graham and Willshaw [64] compare the CA3 region of the rat hippocampus, widely believed to be associated with learning and short-term memory, to their

winner-takes-all associative net model of heteroassociative memory, and conclude that CA3 can store on the order of thousands of patterns with high information efficiency. Efficiency is defined as the amount of information that can be retrieved from the memory to the amount of storage available.

Wolff [191] suggests that the information storage capacity of the brain lies approximately between 3000Mb and 30,000Mb, and that it may be possible to store all the information that a human needs to know without having to resort to hierarchies of cell assemblies. However, he makes many assumptions, for example, that every neuron in the brain is involved in memory storage. Clearly, this is not the case: neurons are needed for control of basic biological functions, such as breathing and homeostasis. Sakurai's argument is more compelling in that it concludes that, even if every neuron in the brain were involved in information storage, there still would not be enough neurons if they did not form hierarchies. This, incidentally, is also the argument against the presence of "grandmother cells" in the brain [6], *i.e.* concepts such as a person's grandmother being represented in single dedicated cells: There simply are not enough neurons in the brain to accommodate all the concepts. Similarly, it is impossible to have a "gnostic unit", a group of cells dedicated to representing a single concept, [106] for all possible concepts, although such groups may exist for commonly accessed concepts.

Most unconvincing of all is Wolff's statement that some sort of information coding (equivalent to replacing repeated patterns in a bit string) is necessary in order to fit all the information in the brain. He proposes that this coding takes the form of "markers" in the cell assemblies that link them. Surely, such coding is equivalent to some sort of hierarchical mechanism, and yet Wolff argues that no hierarchy is necessary. Wolff proposes a framework for cognition and perception called *information compression by multiple alignment, unification and search* (ICMAUS), in which the concept of the cell assembly is adapted so that no neuron is present in more than one assembly, and yet cell assemblies may contain 'references' or 'codes', in the form of neurons linking to other assemblies, that link cell assemblies together [192].

Common sense indicates that the large rate of cell death in the brain also suggests the necessity for coding information in complex overlapping hierarchies of cells. If

concepts were stored in dedicated grandmother cells, then the death of that cell would result in the concept being forgotten completely. While humans are often unable to recall things, such a process cannot be as simple as suggested by the grandmother cell hypothesis. Things apparently forgotten are often remembered later. If all concepts were stored in terms of single cells, then the death of the appropriate neuron might result in a person instantaneously losing the power of speech, or forgetting common words such as "the" or "a". Since a large number of brain cells do die over the course of a human lifetime, and yet such fundamental concepts in the brain do generally remain intact (degenerative conditions such as Alzheimer's disease notwithstanding), concepts must be encoded in groups of neurons, each of which contains many cells.

Pulvermüller [146] demonstrates how MRI scans showing activity in widely distributed areas of the brain (such as Wernicke's and Broca's area, associated with speech, and the visual cortex at the back of the brain) can be explained in terms of cell assemblies that link these areas. However, MRI cannot show the activity in individual neurons. The finest resolution that MRI can achieve is groups of approximately 100,000 neurons. Pulvermüller [145] proposes that each word stored in the brain, or more generally each item stored in memory, is represented by a single cell assembly. He postulates that simultaneous activity in these different brain areas, for instance, when seeing a hammer and saying the word "hammer", would encourage connections to grow between them. This concept of distributed representation by means of transcortical cell assemblies does presuppose that axons of neurons are positioned so that connections can develop between two widely spaced areas, but this is not unfeasible, since axons can be surprisingly long (several centimetres) given the overall size of the brain cell [7]. Evidence of such connections forming in monkey brains has been found by Rizzolatti and Arbib [149]. If his theory turns out to be the case, this wide distribution of cell assemblies would make them very hard to find - one would have to trace the axons of neurons throughout the brain. Some distributed activity has been found by Gray *et al.* [65] who detected synchronous neural firing via electrodes spaced 7mm apart in Brodmann's Area 17. Indirect evidence comes from various behavioural experiments in which adult subjects asked to reach for a known object often look at different objects with similar sounding names [176].

Braitenberg [24] proposes a mechanism whereby the visual cortex of the human brain may use cell assemblies to detect lines in the field of view of specific orientations. He does not describe any experiments which demonstrate this, but claims that his theory does explain the results found by others [16, 87]. Haykin [73] even goes as far as to suggest a neurochemical mechanism by which the synaptic connections may strengthen when the cells on each end of the synapse are both active.

Bliss and Lomo [17] first demonstrated that LTP and LTD occurred in the hippocampus. They were the researchers who first coined the term "long-term potentiation", and who demonstrated that the connections between neurons could be strengthened artifically by applying stimulation to them. They found that the strength of test pulses applied to the major pathway entering the hippocampus was increased after applying brief, high frequency trains of pulses to the same pathway. Furthermore, they discovered that this effect was very long lasting. Sommer *et al.* [174] show that a bidirectional associative memory does indeed form a good model of the long-range connections within the human cortex, which are known to have many feedback connections, and that bidirectional retrieval aids recall and gives a high degree of fault tolerance.

Just as obtaining direct proof in favour of cell assemblies is next to impossible, so is obtaining evidence to the contrary. Those that try to prove that cell assemblies do not exist in the brain are faced with the difficult task of demonstrating a totally negative case. However much evidence accumulates that certain areas of the brain do not contain cell assemblies, their proponents can always point to other areas of the brain where they may be present.

Various statistical methods for detecting the presence of cell assemblies have been proposed. Martignon *et al.* [123] propose three methods that may indicate their presence, based on higher order temporal patterns, all based on the correlations of firing patterns of the neurons involved. A similar method was used by Hetherington and Shapiro [78] who used Pearson's Correlation Coefficient as a simple measure to detect the presence of persistent patterns of activity in simulated cell assemblies. Such a method worked well in their experiments as cell activity tended to persist for a long time, *i.e.* cell activity in a persistent cell assembly tended to dwindle over

time rather than dying out and being reactivated, according to the traditional Hebb model. For models in which such reactivation occurs, Fourier analysis may prove to be a valuable tool in detecting repeating patterns of cell activity. As things currently stand, correlated temporal firing patterns, *i.e.* syn-firing for considerable numbers of neurons, is generally accepted as the best evidence for cell assemblies [55].

Objections to the cell assembly concept are easily tackled. Some, like Milner [134] claim that cell assemblies would not be capable of forming hierarchies and complex structures because the connections between assemblies would essentially be the same as those within them, *i.e.* that interassembly associations would lead to assemblies merging with each other. This assumes that connections between assemblies are identical in strength to those within assemblies. One possible refutation of this has been proposed by Levy *et al.* [115], who point out that cortico-cortical dendrites of pyramidal neurons in the cortex generally fall into two categories. Most complex processing takes place in the cortex, which is divided into six distinct layers each approximately 1mm thick. The *apical dendrites* generally connect the neurons in lower layers to cortical layers closer to the outer edge of the brain. The basal dendrites generally connect neurons to those within the same layer. This is illustrated in figure 1.4. Levy *et al.* propose that the basal dendrites connect neurons to those within the same cell assembly, whereas apical dendrites represent connections between cell assemblies. This does suggest that cell assemblies may be well connected within layers of the cortex but less so between layers.

Such an explanation is generally not necessary, however, as Hebbian learning also includes the concept of weakening connections. Hebb's theory would therefore suggest that connections within an assembly representing a single concept are therefore strong, whereas those between assemblies are relatively weak, since cells within the same assembly co-fire more often than cells in different assemblies.

Milner also wonders why parts of concepts, such as 'doors' and 'windows', are not subsumed into the assembly representing the whole concept, such as 'house'. These are essentially two sides of the same question. It is one of the aims of this thesis to refute these objections by demonstrating that hierarchies of assemblies can easily exist within a network, and that a compound assembly does not automatically

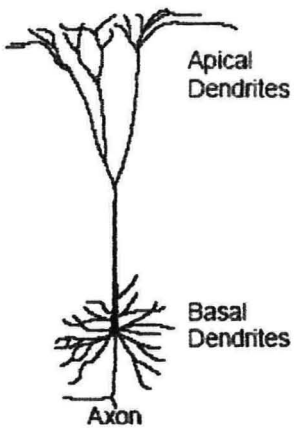


Figure 1.4: Cortico-cortical dendrites on pyramidal neurons in the cortex tend to be apical, connecting the cell to outer cortical layers, or basal, connecting the neuron to those in the same layer. Reproduced from [177].

absorb its components. Milner seems to have modified his view subsequently, and now believes that the brain may contain some variation on Hebb's model of the cell assembly [135].

One objection to the concept of cell assemblies applies to their existence in the visual cortex, and is known as the *superposition problem*, illustrated by figure 1.5 [55]. Layers of cells in brain areas V1 and V2 form *retinotopic maps* in which neighbouring cells in the cortex represent neighbouring areas in the visual field. Cell assemblies are proposed as distributed structures, and any object in the visual field ignites several cell assemblies (representing colour, texture etc.) It is likely that active assemblies will superimpose in the cortex, especially for objects that are close or overlapping in the visual field, and yet the brain has no problem in correctly linking the features to the objects. This is essentially a variation on the variable binding problem [133]. The superposition problem ceases to be an objection to the presence of cell assemblies in the brain if a possible solution for it can be proposed, and such a solution does exist, in the form of *temporal binding*, in which neurons in cell assemblies fire at certain frequencies to indicate their presence in any particular group of features.

The majority of Hebb's detractors, such as Amit [4], dismiss his theory by simply claiming that it is not necessary to explain neurological behaviour. The mechanism

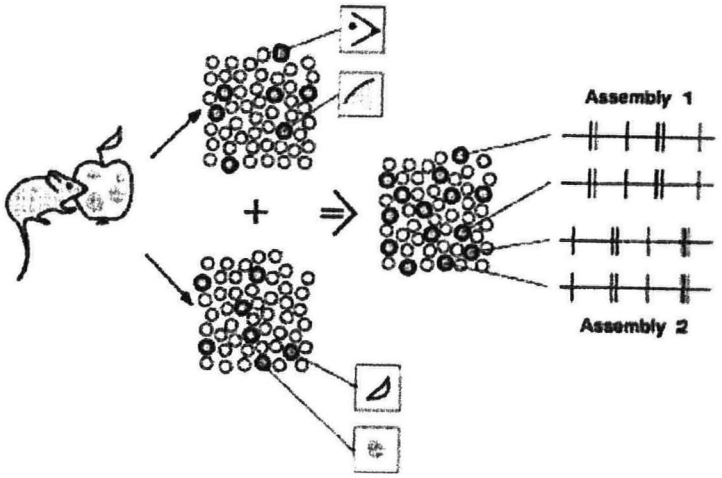


Figure 1.5: The superposition problem, reproduced from [55]. The image consists of two overlapping objects, a mouse and an apple, with the result that all the cell assemblies corresponding to the different features at the point of overlap are ignited. These produce intersecting patterns of activity in nearby cells. The theory of temporal binding proposes that assemblies representing features to be grouped share a common firing pattern.

that Amit then goes on to propose for short-term memory is a network similar to a Hopfield net [84], which moves from one stable state (an attractor) to another stable state. Such stable states are defined by the weights from each node in the network to the others, equivalent to the synaptic strengths between neurons. However, Amit does not specify exactly how these connections could be set up in the brain, and hence does not dismiss the possibility that they may be established by some sort of Hebbian learning. Some researchers, such as Hanson [71] accept that Hebbian learning may be applicable to single cells, but dispute that Hebb's conclusions can be expanded to cover entire networks. It is not the concept of cell assemblies in particular that Hanson is objecting to, but the extension of simple rules such as Hebbian learning to cover entire networks. However, the concept of cell assemblies results directly from an extension of Hebbian learning, so Hanson's criticism may be taken as a criticism of the cell assembly theory.

One valid criticism levelled at the cell assembly theory is that the sparse connectivity of the brain does not lend itself to the formation of cell assemblies. Since the brain contains approximately $10^{10}$ cells, and each neuron has only a few thousand connections at most, each brain cell can only have connections to a tiny fraction of the cells within the brain. Braitenberg and Shuz [25] have carried out a study of the mouse cortex, in which they found that the probability of any pyramidal neuron being connected to a nearby pyramidal neuron was approximately 1 in 50. This does seem rather low to permit cell assembly formation. Palm [139] suggests that neurons in the cortex may be partitioned into modules. Within these modules the neurons have a high connectivity (perhaps a 50% chance of being connected to any nearby neuron), but connections are sparse between modules. There is an alternative interpretation to Braitenberg and Shuz's study, namely that the pyramidal cells do form cell assemblies, but that these assemblies are not composed of spatially contiguous neurons, *i.e.* the assemblies are massively interleaved [63]. Braitenberg and Shuz estimated that each pyramidal cell had approximately 4000 connections, almost all to different neurons. The Braitenberg and Shuz study therefore only undermines the cell assembly theory if one assumes that assemblies must be composed of adjacent neurons.

Although Hebb's theory is more than fifty years old, it has survived the test of time so far. However, it is only within the last twenty years or so that we have had the technology to investigate it thoroughly. This has resulted in a fair amount of circumstantial evidence in favour of the theory and some theories which contradict it. It is my opinion that the weight of the evidence, such as it is, is in favour of both Hebbian learning and the presence of cell assemblies. However, until and unless direct evidence is discovered which proves that such self-excitatory loops exist within the brain, the idea of Hebbian learning and cell assemblies will remain just a theory, open to both support and dispute.

Whether cell assemblies exist in the brain or not, their simulation in computer programs opens an interesting avenue of research. They have already shown that they can exhibit complex behaviour, including the properties predicted by Hebb for biological cell assemblies. Most researchers who investigate simulated cell assemblies do so simply to determine what information processing properties they have, but inevitably, when such simulations succeed, they implicitly lend support to Hebb's theory. As networks of simulated cell assemblies become more complex, they will take on higher functioning capabilities, and it will be reasonable to refer to them as modelling brain processes. For this reason, I believe that a brief discussion of modelling brain function is justified.

## 1.2 Why do we model the brain?

One of the most powerful reasons for studying simulations of Cell Assemblies is that they are the only widely recognised neural network architecture that are based closely upon neural structures that we believe are present in the brain. The whole foundation for cell assemblies is the 1949 book by Hebb [74], one of the leading neurologists of his day. Although Hebb did not specify equations for his proposed learning, the equations generally used are based loosely on more modern neurological research, such as that of Hubel and Wiesel [86, 88]. Already models of brain function, such as the motor control system [40] and the cortex basal ganglia system [79], are being implemented based entirely upon cell assemblies. It therefore makes sense to discuss

briefly the reasons for simulating brain function in computer programs.

The most obvious answer to this is that models of the brain give some insight as to how it works. The simplistic approach to this problem is to view the brain as a single immense and densely interconnected network of neurons. However, the vast numbers of neurons in the brain, each of which forms many thousands of connections, makes an exhaustive model of the brain impractical.

Models of the brain or mind allow us to concentrate our attention on those characteristics of the brain that are salient for any line of research, and to ignore others. They are embedded in assumptions about our thinking [180]. A brain model would be a source against which theories could be tested [158].

A model of (small areas of) the brain provides a useful test-bed for neurological experiments that would otherwise be impractical. It is estimated, for example, that IBM's proposed model of a complete cortical column [47] will be able to complete experiments in seconds that would take days if carried out using a real human brain. The development of the brain from before birth to old age could be surveyed in a tiny fraction of a human lifespan, and diseases and disorders simulated by artificially lesioning connections in the model. Furthermore, simulations of brains and minds can be analysed thoroughly and "dissected" in a way that would never be acceptable with human subjects. At the other end of the scale from IBM's model are simulations that contain only a few cells and yet have produced useful results and have cast light on pathological neural conditions, such as [124] in which a model of lexical retrieval simulated the errors produced by a brain-damaged patient.

It should be pointed out that the validity of the experiments described in this thesis do not rely on the presence of cell assemblies in the brain. I clearly demonstrate that they have a useful function regardless of Hebb's theory. However, the fact that they are modelled directly on our best understanding of the brain, the most powerful neural architecture known, provides a good reason for researching into them.

## 1.3 Connectionism vs. Symbolic AI

Cell assemblies form part of a vast area of research called *neural networks* or *sub-symbolic AI.* In the very early days of AI, the subject divided into two fundamental strands, symbolic AI and sub-symbolic AI, also known as connectionism. Connectionist systems were loosely inspired by the workings of the human brain, which consists of many types of neurons that pass electrochemical signals to each other. Symbolic AI, the older of the two approaches, comprises the storing of information in computers in the form of discrete and easily identifiable items, such as simple variables. Processing of these pieces of information is carried out by means of rules, each of which has a specific purpose. An example of such a rule, drawn from an expert system designed to identify animals from a given set of features, is shown below:

```
IF animal_can_fly AND animal_is_mammal
THEN animal_is_bat
```

The purpose of this rule is obvious simply by reading it, as are the pieces of information that it needs to operate. Of course, expert systems are best suited to problems which encode easily in the form of rules, such as diagnosis of diseases or problems in mechanical equipment. Many problems in the real world, such as speech or facial recognition, are less suited to this rule-based strategy. In such problems, the expert knowledge required is harder to state explicitly, and the more numerical approach offered by connectionism is preferable.

Connectionism adopts a completely different approach to processing information. Information is distributed throughout neural networks in the form of weighted connections between homogenous components. It is difficult to pinpoint particular concepts within the network. Furthermore, activity is propagated throughout the network through the global application of the same rule, and hence information processing must also be encoded in the weighted connections. Consistent global behaviour emerges from local operations. There is no reason to suppose that simply because connectionist models are inspired by the brain that all models of brain processing must be connectionist in nature. There are many cognitive science models that are entire symbolic in nature (see [172]), or even a mixture of the two, such as Wolfe's

Guided Search 2.0 model of attention [190]. Indeed, between the late 1960s and the mid-1980s, almost all research in the field of cognitive science was conducted using symbolic AI [81].

Connectionism has its roots in the 1950s and 60s with systems such as Selfridge's Pandemonium [159]. The early connectionists did make over-ambitious claims for their architectures. Rosenblatt [151], for instance, claimed that his perceptrons were capable in principle of having original ideas, and McCulloch and Pitts [128], credited with developing the first model of the neuron, stated that "specification of the net would contribute all that could be achieved in [psychology]". Inevitably, there was a backlash from the Symbolic AI community. Connectionism was dealt a blow by Minsky and Papert [136] who showed that the basic element of neural networks at the time, the single-layer perceptron, could not discriminate between patterns that were not linearly separable. Within the last twenty years or so, the development of more powerful architectures such as the Multi-Layer Perceptron and the Mixture-of-Experts [163, 97] has given rise to a resurgence in interest in connectionist architectures, to the extent that it is generally accepted that connectionism has permanent place in the armoury of AI tools.

Each approach has its advantages and disadvantages. Neural networks have a certain degree of redundancy insofar as it is possible to lesion a proportion of the connections between cells and still the network will perform its designated function with only a slight decrease in performance. This property of *graceful degradation* is not shared by symbolic AI programs, in which missing out any particular rule or variable can lead to complete failure. Neural hardware, such as the brain itself, demonstrates graceful degradation - this must be the case since human brain function continues in spite of the fact that we each lose countless brain cells every day - but to a certain extent this advantage is lost when a connectionist net is simulated on a serial computer. The main advantage of symbolic AI is that the functioning of a program can be easily analysed. It is a fairly straightforward matter to determine which part of a program performs which function. This is not the case with neural networks, whose functions are distributed amongst all the neurons and leads to connectionism often being referred to as *distributed processing*. It can be difficult, if

not impossible, to determine the function of any particular cell in a neural network. This is another facet of the redundancy aspect: Redundancy provides resistance to catastrophic breakdown, but it does make analysis difficult.

Certainly connectionist nets cannot (as yet) provide all the functions present in symbolic AI. Fodor and Pylyshyn [52] suggest that neural nets cannot be used to form general rules that draw inferences. For example, in symbolic logic, one can deduce from the rule "$A$ is true and $B$ is true" that "$A$ is true". Such a rule translates easily into "$C$ is true and $D$ is true" from which we can deduce that "$C$ is true". However, many researchers do not accept their arguments [171] and some even claim to have developed connectionist nets that support systematic operations [30].

One area in which connectionism does appear to have a distinct advantage over symbolic logic is in the representation of uncertainty. In symbolic AI, uncertainty arises when one is not sure to what extent any proposition is true. For instance, in an expert system designed to diagnose disease, the degree to which a patient displays a particular symptom may be uncertain. Many formalisms have been developed to accommodate uncertainty, the most common of which are Bayesian Reasoning [44], the Dempster-Shafer Formula [160, 196], the MYCIN calculus [26, 56] and Fuzzy Logic [195], all of which are summarised in [21]. Much controversy exists as to which formulism is best, mainly between the proponents of Bayesian Reasoning, who point out its mathematical rigour, and the non-Bayesians who point out the ease of use of their own approaches [116, 32].

Connectionism avoids this problem automatically. Uncertainty is built into its processing, as inputs to connectionist nets are continuous-valued, and this continuity is maintained throughout the nets. A good example is the network known as "Jets and Sharks" [126, 11]. This deals with the characteristics, such as education level, marital status etc., of a group of people who are members of the two eponymous gangs. The network is capable of an associative database, to which various characteristics are applied. The signal strength of the nodes of each gang member then indicates the degree to which that member matches each of the applied characteristics. Figure 1.6 shows the result of applying an external signal to the network node representing "Shark", *i.e.* a query to find all the members of the Shark gang. The activity

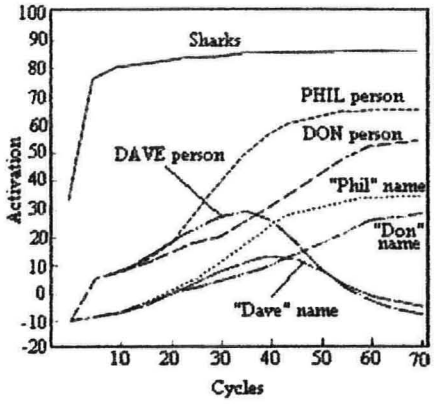


Figure 1.6: A typical query applied to the "Jets and Sharks" network.

level of cells representing various members gradually increases, together with those representing other characteristics commonly shared by several members of the Shark gang.

Clearly the network has decided that both Phil and Don meet the criterion "Shark" well, but less so Dave. The inherent uncertainty in the output is obvious from the figure without the necessity to impose an artificial uncertainty formulism.

While both the symbolic and connectionist approaches to AI have their advantages, each has its drawbacks also. While connectionism allows us to solve problems which do not lend themselves easily to solving by a series of rules, it is not easy to determine how the knowledge has been stored in the network. Connectionist networks are essentially parallel in nature, a fact that should lead to high processing speeds, but this advantage disappears when the connectionist architectures are simulated in serial computers, as the vast majority of them are. In spite of attempts to discredit connectionism, it has fought its way back to earn a place as a useful tool in any AI programmer's toolbox, and researchers are by no means close to discovering its limits.

Some researchers claim that cell assemblies can be classified both as connectionist and as symbolic architectures [62]. They consider the question of how cell assemblies store knowledge. Certainly, they are connectionist, in the sense that they are based on cell-like structures. However, like conventional symbolic AI systems, they are modular in nature, with smaller modules contributing to hierarchies of larger ones.

## 1.4 Objectives

The main objectives of this PhD project are summarized by the sub-headings in this section. By "cell assembly" I mean cell assemblies simulated in a network of cells on a computer.

### 1.4.1 Establishment of a cell assembly in a network of cells

The first task is to establish that cell assemblies do indeed form in a network of uncommitted cells. A primitive of cells should be created with a given number of connections between them of random strengths and destinations. A simple learning rule can be implemented that allows connection strengths between cells to alter. By stimulating parts of this network appropriately, it is hoped that cell assemblies form spontaneously and that these cell assemblies demonstrate all the properties believed to apply to cell assemblies in the brain [154]. It is important that adaptation of connections be carried out based purely on a local metric. Cells in the brain can have no global "knowledge" of the problem to be solved or strengths of distant connections. The experiments must show that global order can be imposed on an initially random system through the application of a few simple rules. To a certain extent, these experiments will help to dispel criticisms of the cell assembly argument by researchers such as Milner [134], who cannot appreciate how component cell assemblies can contribute to compound assemblies without being subsumed by them. This is not a main aim, since these criticisms carry little weight, and the existence of cell assemblies in the brain is now widely accepted. However, it will validate such concepts in simulated cell assemblies.

### 1.4.2 Investigation of properties of cell assemblies

If cell assemblies exist in the brain, then they are capable of a variety of behaviours that promote the storage and adaptation of information. The brain forgets information, so it is quite possible that cell assemblies may disband. Similarly, information may be stored in very generalised terms to begin with and become more specialised later, such as learning the concept of "animal", followed by specific animals such

as dogs, then breeds of dog. One possible mechanism by which this may occur is fractioning (splitting) of cell assemblies into subassemblies, which then recruit uncommitted cells on their boundaries. Experiments are carried out that investigate whether forgetting, fractioning and recruitment at the boundaries of assemblies can be made to take place.

The cells from which cell assemblies simulated on a computer are created can only ever be approximations of biological neurons. As such, they are modelled by a number of equations. These equations are in turn based on a number of parameters. The thesis investigates the relationships between these parameters that hold for networks that display acceptable behaviour.

### 1.4.3 Establishment of correlates between simulated cell assemblies and those in the brain

Although a single cell assembly lacks the sophistication of other simulations in cognitive psychology, I feel that is fair to describe the experiments described in this thesis in terms of cognitive psychology. Some neuronal functions have indeed been described as the result of cell assembly formation. A notable example is the setting up of cortical hypercolumns [24]. Of the 305 connection pathways between segregated areas in the brain, more than 80% of them have fibres running in both directions [49]. This high degree of feedback has prompted some researchers to claim that cell assemblies form the basis of all brain activity.

While my research cannot make any conclusive statement about how cell assemblies may or may not operate in the brain, it is hoped that the simulations of assemblies may throw some light on the theories of brain activity and suggest possible avenues of research.

### 1.4.4 Determination of the storage capacity of cell assembly networks

It is generally accepted that the number of neurons in the cortex would be insufficient to store the number of memories required of it if each memory were assigned to a

unique neuron. The storing of that many memories requires that individual neurons be involved in the storage of more than one memory. This implies some sort of hierarchical structure in which groups of neurons co-operate. A similar approach can be taken to simulated cell assemblies, *i.e.* the number of assemblies that can be stored in a network is not necessarily proportional to the number of cells in the network. Of course, the number of connections between cells in the assemblies, whether synapses connecting biological neurons or simulated connections in a computer program, must be sufficient to allow the hierarchies to be set up.

Chapter 2 on page 31 gives brief details of some of the approaches that different researchers have taken to determine the storage capacities of Hopfield nets, all of which make slightly different assumptions, and which produce strikingly different capacity formulae as a result. There is, however, yet no body of research that performs the same function for cell assemblies. A large part of the thesis investigates what structures can be set up in a network in order to increase the storage capacity of that network.

## 1.5 Summary

Cell Assemblies promise to be a powerful tool both for modelling the brain and for developing more powerful architectures for simulation on a computer. If Hebb is right, and cell assemblies are indeed present in the brain, then they may well be capable of sophisticated information processing. The work described in this thesis does not attempt such sophisticated computation, but it may be possible to proceed from simple information storage to, for instance, sequential information processing and even decision making. The experiments described show increasing complexity from the implementation of single cell assemblies, via small networks of assemblies, to large networks containing even larger numbers of assemblies.

Chapter 2 describes the important literature in the field of cell assemblies. It also gives a brief overview of some of the ancillary topics that have affected the experiments described in later chapters, such as genetic algorithms. These are not investigated in great detail since they are not central to the experiments. Chapter 3 describes in

mathematical terms the components of all the simulations described in the thesis. It gives the equations that govern the behaviour of the cells in the networks, those that implement the Hebbian learning used, and the details of the genetic algorithm used to determine acceptable values for the parameters that control the model's behaviour. This model is then used to establish the validity of small networks of cell assemblies in chapter 4, in which the complexity is increased gradually from one isolated assembly to a hierarchy of six assemblies. Chapter 5 then extends this work to large networks, containing an indefinite number of primitives. The main interest in chapter 5 is the storage capacity of the network, *i.e.* the number of patterns that can be stored and reliably recalled, as the size of the network increases. At this point, comparisons with large Hopfield nets (section 2.3.1) are made. The final chapter, chapter 6, draws together all the threads from the previous ones into the relevant conclusions, and makes suggestions for future research.

## 1.6 Terminology

In the following chapters, the word *neuron* refers to a brain cell, and the word *cell* is used to refer to its simulation in a computer. One exception to this is the term "lucky neuron", which applies to certain critical cells in experiments. The connections between cells in simulations are often referred to as *synapses* after their biological equivalent.

All experiments are labelled with the section number in which they are described.

# Chapter 2

# Literature Review

The concept of the cell assembly is just over half a century old and has become established as a partial explanation of how thought and memory are encoded in the neural structure of the brain. This chapter puts simulations of cell assemblies into the context of other neural architectures. It also gives a brief introduction to various topics that have some bearing on the experiments described later in the thesis. Some of these, such as the discussion on the capacity of associative memories, are dealt with in some detail.

## 2.1 Modelling brain cells

The basic structure of neurons has already been discovered. Although there are many types of brain cell, they can be approximated by the diagram shown in figure 2.1.

The cell consists of a cell body, or soma. Electrical signals are fed into the cell body via thin tree-like structures called dendrites. The cell is usually inactive, but

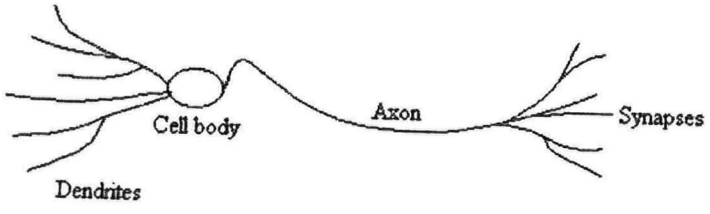


Figure 2.1: An approximation to the structure of a brain cell.

when a sufficient quantity of signal has been received from the dendrites, the cell becomes active, it fires, and a further signal is sent via a long structure called the axon to synapses. These are connections between the axon of one brain cell and the dendrites of others. The signal is transmitted from the axon to the dendrites via chemicals called neurotransmitters. When the cell has fired, it usually returns to the inactive state. This return is referred to as the spike latency.

Synaptic inputs operate by increasing or decreasing the conductance of the post-synaptic membrane. This implements the strength of the synaptic connection. When most neurons fire they stimulate the cells to which they connect to build up the electrical charge in their cell bodies. These cells, based on NMDA, are excitatory [166]. NMDA is N-methyl-D-aspartic acid, an amino acid derivative acting as specific agonist at the appropriate receptor on synapses. A small proportion of neurons, based on GABA, seem to have the opposite effect. When they fire, they decrease the activity in the cells with which they come in contact. Such cells are said to be inhibitory. GABA is Gamma-aminobutyric acid, a neurotransmitter that acts on any of three receptors on synapses, termed $GABA_A$, $GABA_B$ and $GABA_C$.

In the brain, neurons fire the instant that their internal conditions permit it, without reference to other cells. Furthermore, synaptic connections operate at different speeds, typical reaction times varying from 1 ms to 100 ms [156]. These effects taken together are termed asynchronous firing. Clearly, when cells are modelled in a serial computer, it is difficult to model asynchronous firing perfectly, as the simulation program can only handle one cell at a time. Simulations generally involve processing all cells in the network serially.

All connectionist systems involve modelling neurons to a greater or lesser degree, as they are inspired by the brain. Architectures such as the Multi-Layer Perceptron (MLP) contain cells that implement very simple behaviour, *i.e.* the weighted connections together with a simple transfer function from input to output. Simplicity improves program speed, and such simple cells suffice for the tasks to which MLPs are applied. As models become more complex, they take into account the internal workings of neurons, for example, simulating the currents of sodium and potassium ions [111]. Such models often become unwieldy, such as the family of neuronal models

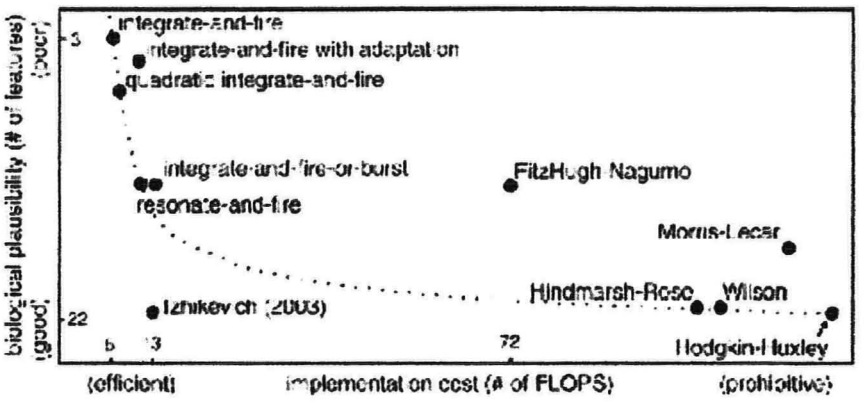


Figure 2.2: The more biologically plausible a neuronal model is, the more floating point operations (FLOPS) it requires to implement. (Reproduced from [96])

generally referred to as Hodgkin-Huxley models. These involve only four equations [80], but are based on dozens of parameters, each of which needs to be tuned individually for any given simulation. Izhikevich [96] lists no fewer than twenty different models of varying complexity, recommending the Hodgkin-Huxley model for detailed and accurate modelling of the neuron when only a few cells are to be modelled and the quadratic Integrate and Fire (I&F) model for large numbers of simulated neurons. Inevitably there is always a trade-off between the number of mathematical operations involved in the simulation of neurons and the biological plausibility, as shown in figure 2.2.

Experiments described in this thesis often involve huge numbers of cells and many thousands of iterations. The I&F model is the simplest model that is generally recognised as simulating neuronal function to any degree above the basic level found in architectures such as MLPs [96]. For this reason, the model used is based on the I&F model, but incorporates the additional functions of leaky integration and fatigue as described in chapter 3 on page 53. Essentially cells in the model accumulate activity through input connections, a fraction of which leaks away as time progresses. Cells also have the property of fatigue that prevents them firing indefinitely in the presence of continual external activation. Although it sacrifices biological plausibility for efficiency, it has proved satisfactory in various cell assembly experiments such as [91] and [93].

## 2.2 Cell assemblies as neural networks

This section briefly discusses the position that cell assemblies occupy in the hierarchy of artificial neural networks such as multi-layer perceptrons [117] and Kohonen nets [104]. Although it is still a matter of debate whether cell assemblies exist in the brain, they may well prove to be a useful tool in artificial simulations of neural networks.

Cell assemblies form their own representation of the input data. In this respect they are self-organising or unsupervised, like Kohonen's self-organising topographical maps (SOTMs) [105]. Whether this is an advantage or not is debatable: on the plus side, there is no need to label training data, which allows the cell assemblies to extract the important features from the data for themselves. In many experiments networks are created containing cells with randomly assigned connections, both in strength and destination. In these experiments, cell assemblies form spontaneously as a result of external stimulation. In these experiments, the programmer has little control on how the cell assemblies form[1] - it is a hit and miss process. Cell assemblies are initialised with the minimum number of the parameters specified by the user - the weights between connections are set randomly and the destination of the connections set randomly as well. The connection strengths between nodes in a SOTM are set randomly but the nodes are only connected to neighbouring nodes in the grid. This implies that SOTM will produce more predictable grouping of data inputs, whereas the cell assemblies are much less predictable as regards their outcomes.

The fact that SOTMs undergo unsupervised training means that they do not necessarily classify data in the way that a human would find intuitive. In general, they tend to be used as vector quantizers, *i.e.* they cluster data items and then pass the result on to other components in a modular neural network (see [9] and [193] for examples). The same could apply to cell assemblies, of course. The experiments described in this thesis describe situations in which the topologies of networks were carefully controlled. In more complex situations, the final topology of the network may well be less tightly restricted, in which case some external system will be required

[1]Experiments described in this thesis use networks in which the positions of cell assemblies are predetermined by predetermining connection strengths and/or restricting external stimulation to specific areas of the network. Cell assemblies in such experiments are a lot more predictable.

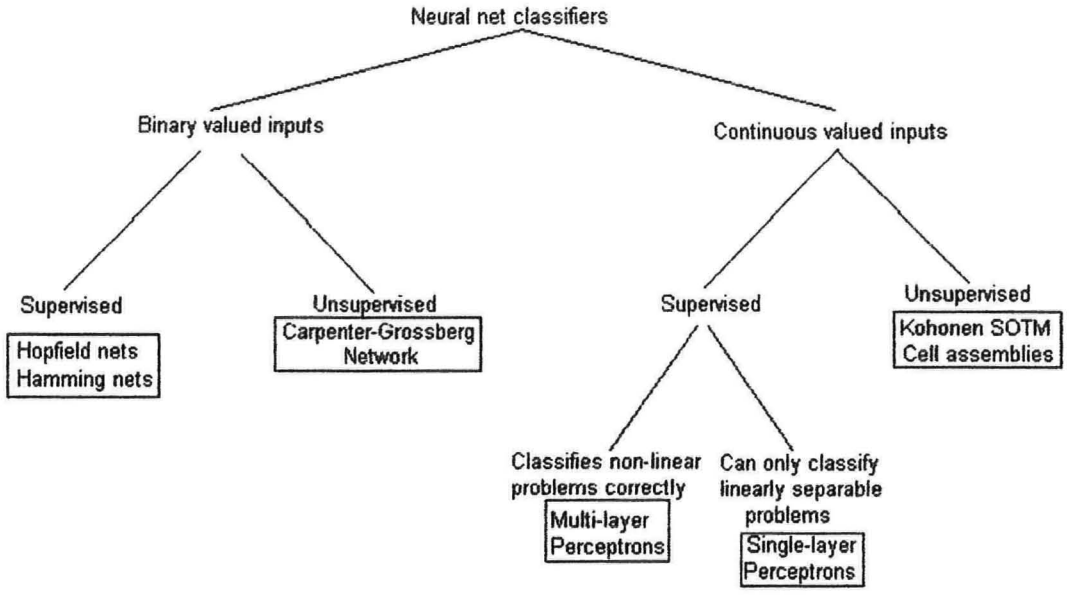


Figure 2.3: A taxonomy of neural net classifiers, showing the position of cell assemblies.

to "interpret" the result.

Cell assemblies undergo adaptive training. This means that the network weights are adapted during training, unlike Hopfield nets [84] and Hamming nets which tend to use fixed weights. In this respect they are similar to multi-layer perceptrons. Figure 2.3 shows a generally accepted taxonomy of neural network architectures [117] to which cell assemblies have been added. The essential differences between cell assemblies and Hopfield nets are explained in section 2.3.1 on page 39. In spite of the fact that cell assemblies are more closely related to Hopfield nets than they are to SOTMs, they appear distant from them in figure 2.3 due to the fact that Hopfield nets always deal with binary inputs and outputs. Although simulated cell assemblies are implemented in such a way that their stimulus is continuously valued, the definition of cell assemblies has not yet been cast in stone, so that there is no reason why they could not be given binary inputs. For this reason, I maintain that Hopfield nets are the closest relative that cell assemblies have in the set of standard neural architectures, and I concentrate on comparing cell assemblies to Hopfield nets rather than to SOTMs.

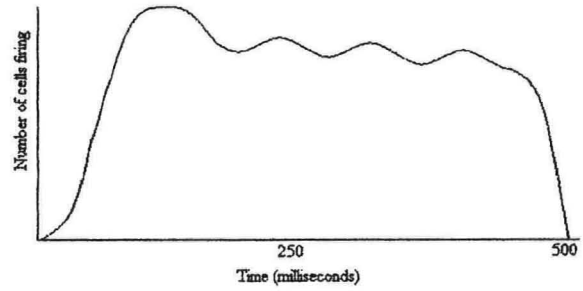


Figure 2.4: The TRACE model of cell assembly activity. The vertical axis shows the relative activity of the assembly in arbitrary units.

Kaplan *et al.* [98] created a mathematical simulation of a cell assembly that fits one possible model of their behaviour. TRACE simulates not the behaviour of individual cells in a network but the total activity level in the entire cell assembly. This is roughly analogous to the number of cells that fire at any point in time. According to the TRACE model, activation in cell assemblies follows the pattern shown in figure 2.4, in which the number of cells firing increases rapidly. Firing reaches a peak as cells begin to fatigue in large numbers, and then reaches a plateau in which the rate at which cells recover from fatigue is approximately equal to the rate at which fatigue prevents cells from firing. There may well be oscillations in the number of cells firing, until the number of cells incapable of firing reaches a critical level and the activity in the cell assembly rapidly drops to zero.

Of course, the TRACE model of cell assembly does not represent the only possible activity pattern. There is no reason, for example, not to assume that the activity of any cell assembly in the brain would persist indefinitely if it were not for the presence of competing cell assemblies that inhibit it. TRACE has been used to model lexical contact during speech perception [53], and has been extended to model activity in multiple cell assemblies (the multiTRACE system, [35, 175]). It is possible to simulate a cell assembly based on discrete cells that matches approximately the activity pattern of TRACE, as shown in figure 2.5. Details of this simulation are given in section 3.4 on page 65.

How many cells should be present in a cell assembly? There is no definitive answer to this question. Fransen *et al.* [54] have demonstrated that a simulated cell assembly

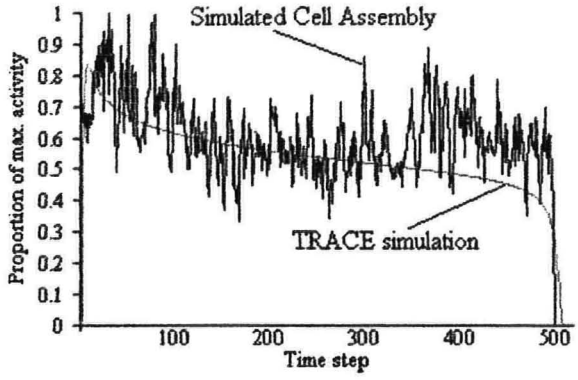


Figure 2.5: The activity level of a network of simulated cells tuned to match the TRACE activity level as closely as possible. The vertical axis shows the activity as a proportion of the maximum possible activity of the network. One time unit on the horizontal axis is approximately equal to a millisecond.

can contain as few as 8 cells, as shown in figure 2.6. Their simulated cell assemblies showed the basic properties of cell assemblies, *i.e.* persistance after removal of the external stimulus (which they term *after-activity*) and completion of the assembly (which they term *pattern completion*). They even demonstrate that one such small assembly can compete with and shut down another (figure 2.6(b)). Few researchers, however, choose to work on such a small scale. Hetherington and Shapiro [78] created assemblies on grids of cells with 18 rows and 20 columns, the CANT system [89] uses grids of cells with 20 rows and 20 columns. It is one of the aims of this thesis to demonstrate that networks containing large numbers of cells have more scope for cell assembly formation than do small networks.

Cell assemblies are gradually being accepted among the useful neural network architectures in common usage. This thesis aims to discover to some extent their capabilities and limitations.

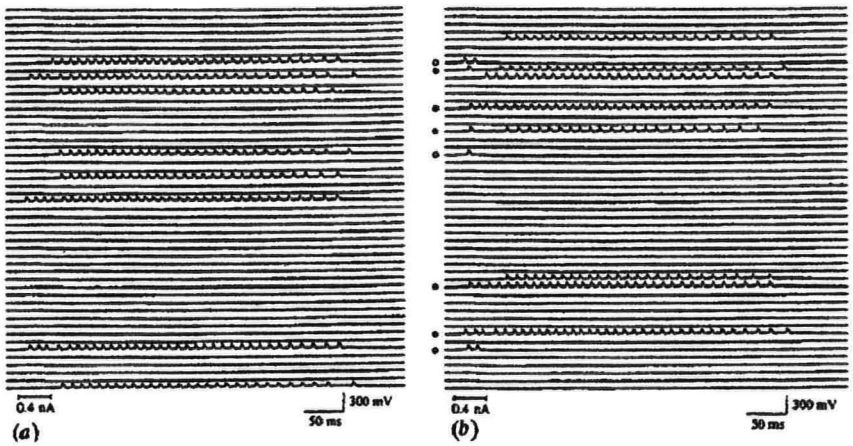


Figure 2.6: Fransen et al. [54] demonstrate basic properties of cell assemblies in small networks of 8 cells each. The activity in (a) persists after the external stimulus shown by the horizontal bar is removed. In (b), cells in the assembly marked by the closed circles compete with and shut down those in the assembly marked by the open circles.

## 2.3 The relationship between Cell Assemblies and Hopfield Nets

Cell assemblies are a connectionist architecture that bear some similarity to Hopfield nets [84, 85] in the way that they operate. Hopfield nets are a standard architecture for associative nets, and are generally used for pattern completion. They consist of a series of cells or nodes with recursive connections to which a test pattern is applied. These connections allow the state of the net, represented by the outputs of the cells at any given point in time, to move from one state to another until stability is reached. They are loosely inspired by the workings of neurons in the brain. Indeed, some researchers have used Hopfield nets to draw conclusions about the brain. Crick and Mitchison noticed that Hopfield nets often become over-loaded with stored memories. From this they made the immense leap to concluding that the human brain avoids this problem by dreaming (the Crick-Mitchison Hypothesis: "We dream in order to forget", later toned down to "We dream in order to reduce fantasy and obsession") [63].

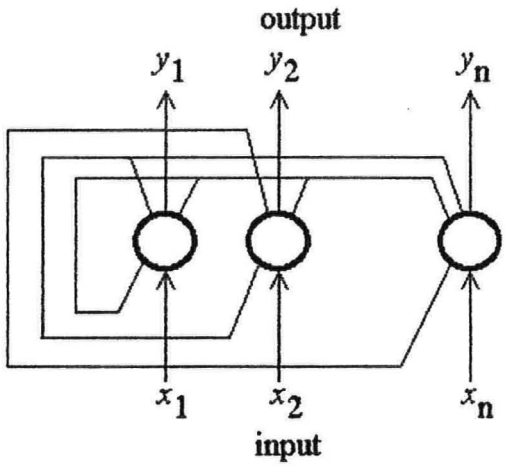


Figure 2.7: A Hopfield net. There are $n$ inputs (termed $x_1$ to $x_n$) and outputs ($y_1$ to $y_n$), and each cell has a recurrent connection to every cell other than itself.

### 2.3.1 Hopfield Nets

Figure 2.7 shows the basic structure of a Hopfield net. It consists of a single layer of $N$ cells, each of which has recurrent connections to all the cells other than itself. The cells take a binary input of $N$ bits and then iterate until they reach a stable state, at which the binary states of the cells is deemed to be the $N$ bit output. The binary states of the net are usually 1 and -1, although versions of the Hopfield net do exist in which the binary states are 1 and 0 [117]. Two advantages that Hopfield nets have over multi-layer perceptrons is that they require only a single training pass to set any particular pattern, and that further patterns can be set at any time without risk of forgetting previously set patterns (as long as the maximum storage capacity of the network is not exceeded).

The algorithm for training and using the Hopfield net is shown in table 2.1 on page 40.

The output of the Hopfield net is read from the $N$ nodes after convergence has been achieved. This output is then matched against the library of patterns on which the net was trained. It has been shown that the preconditions for the net to converge are that weights are symmetric ($t_{ij} = t_{ji}$), which they must be if the standard training algorithm is followed, and nodes are updated asynchronously [84].

1. Assign connection weights between nodes $i$ and $j$ for $1 \leq i, j \leq N$.

$$t_{ij} = \begin{cases} \frac{1}{N} \sum_{s=1}^{M} \xi_i^s \cdot \xi_j^s & \text{for } i \neq j \\ 0 & \text{for } i = j \end{cases} \tag{2.1}$$

The algorithm assumes that the net is being trained with $M$ patterns and contains $N$ nodes. $t_{ij}$ is the weight from node $i$ to node $j$, and $\xi_i^s$ is binary bit $i$ of training pattern $s$. Many researchers, such as [39], refer to this as "one-shot Hebbian learning". Often references, such as [117], omit the $\frac{1}{N}$ scaling term, but including it often improves the ability of the Hopfield net to converge on training patterns rather than on spurious patterns.

2. Apply input.

$$\mu_i(0) = x_i \text{ for } 1 \leq i \leq N \tag{2.2}$$

$\mu_i(t)$ is the output of node $i$ at time $t$. $x_i$, which can be +1 or −1, is element $i$ of the input pattern.

3. Iterate until convergence.

Construct a weighted sum of each output value:

$$h_i = \sum_{j=1}^{N} t_{ij} \mu_j(t) \tag{2.3}$$

Pass this weighted sum through a hard-limiting ("sgn") function, $f_k$ that converts any positive input to +1, with any other input converted to -1:

$$\mu_i(t+1) = f_k(h_i) \tag{2.4}$$

Table 2.1: Hopfield Net Algorithm

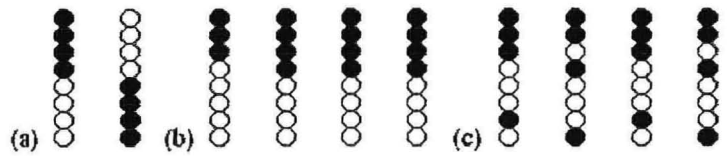


Figure 2.8: A Hopfield pattern trained on two patterns in (a) can show activity that either converges (b) or oscillates (c).

Cell assemblies have important similarities to and differences from Hopfield nets. Like Hopfield nets, they are recurrent and move from one state (as specified by the activation of the cells at any moment) to another. Both cell assemblies and Hopfield nets perform pattern completion, although the exact pattern of cells activated in the cell assembly is less strict than for the Hopfield net. Hopfield nets are designed to converge on stable states, but in many cases they converge to an oscillating pattern [85]. Figure 2.8 shows patterns of activity in a simple Hopfield net trained on two patterns, shown in figure 2.8(a). Figure 2.8(b) shows a test pattern (left most) and the three resulting patterns that follow on subsequent time steps. In this case the activity pattern converges to one of the training patterns in one time step. Figure 2.8(c) shows another test pattern on the left and the three subsequent time steps. In this case, the pattern of activity oscillated indefinitely. Activity in my model does not converge in the same way as in the Hopfield net. In a cell assembly, patterns of activity in the network do not persist unchanged indefinitely from one time step to the next, nor does a cell assembly demonstrate oscillating patterns that manifest themselves perfectly at regular intervals. Convergence is taken to mean that a sufficient number of cells in an assembly fire so that the assembly itself may be considered to be active, even though the exact patterns of activity don't necessarily repeat.

However, there are notable differences between the two types of architectures, that arise from the fact that cells in a cell assembly are based to a greater or lesser extent on neurons in the brain, and include fatigue and decay. Of course, simulated cell assemblies are ill-defined, as each researcher decides on his or her particular variation, whereas Hopfield nets are rigorously defined. The following list gives some of the differences between Hopfield nets, trained using the standard algorithm in table

2.1 on page 40, and cell assemblies as they are believed to exist in the brain.

1. Cells in Hopfield nets are essentially binary in operation, being either active or inactive. Cells in cell assemblies do exhibit binary behaviour insofaras they may or may not fire, but the firing is based on an underlying activity level that is continuous.

2. As stated above, cells in cell assemblies include the features of fatigue and decay, so their activity gradually dwindles in the absence of external stimulation. Cells in Hopfield nets retain their activity indefinitely, unless specifically deactivated by negative connections. Hence, it is more correct to refer to the states into which a cell assembly passes as *pseudo-stable*, rather than stable. This is an important difference since it is this property that allows cell assemblies to move from one state to another, thereby introducing the possibility of sequential processing.

3. A Hopfield net always expresses patterns with the same number of bits as there are cells in the net. This is not necessarily true of cell assemblies. Connections can be set in which patterns of activity can develop independently in different parts of the network and have no influence on each other.

4. Hopfield nets are usually well connected, *i.e.* each cell has a connection to all the other cells. Cell assemblies are sparsely connected.

5. Cells in Hopfield nets can have an excitatory effect on some cells and an inhibitory effect on others. Dale's principle states that neurons can be either excitatory or inhibitory, although recently doubt has been cast upon this (see section 3.1 on page 54).

6. Connections in Hopfield nets are always bi-directional and symmetric, *i.e.* if there is a connection from cell A in a Hopfield net to cell B, then there is automatically a connection of the same strength from node B to node A. There is no such restriction in cell assemblies.

7. The weights of a Hopfield net are adapted only when specific patterns are to be added, either when the network is set up or at some future stage. Weights in a

cell assembly are adapted continually throughout the lifetime of the assembly. This is not to say that simulated cell assemblies cannot be set up with fixed weights. Indeed, experiments in later chapters of this thesis implement just such assemblies, but that synaptic connections within cell assemblies in the brain are always adaptable.

### 2.3.2 Capacity

The capacity of a neural architecture is a measure of the maximum number of patterns that can be stored and retrieved reliably. It is usually expressed as a critical load ($c = \frac{pc}{N}$ where $pc$ is the maximum number of patterns and $N$ is the number of cells in the architecture) [131]. For any architecture with capacity $O(N)$, the critical load for such an architecture converges to a particular value proportional to $N$. Krauth and Opper [108], for example, have shown that the critical load for an autoassociative network has an assymptotic value of $c = 0.833$.

The theoretical maximum for the storage capacity would be $2^N$ if no patterns were to be stored or retrieved, since each cell can be either on or off. However, the purpose of a Hopfield net is to move from a state representing a partial or corrupted pattern to one representing a pure stored pattern, so the net can only function properly if the great majority of the states are unstable, and do not represent trained patterns. Peng and Zhou [141] report early experiments that do indeed suggest that $2^N$ $N$-bit binary patterns can be stored in an associative network with on the order of $N$ cells, while retaining a high level of insensitivity to noise, *i.e.* a network of $N$ cells can store $\frac{2^N}{K}$ patterns. In general, when considering the storage capacity of a network such as a Hopfield net, one considers how many random patterns (*i.e.* each bit has an equal probability of being a 1 or a –1) can be stored and retrieved. Kitano and Aoyagi [101] have pointed out that it is more reasonable to consider storing patterns in which relatively few of the bits are 1. This bears a closer resemblance to real neural systems, which are generally sparsely active, and has a bearing on the large networks of cells described in chapter 5 on page 121 in which only a small proportion of the cells are active at any one point. It is generally considered that the maximum number of patterns that can be stored in a Hopfield net increases in proportion to

$-\frac{1}{a}\ln a$, where $a$ is the proportion of bits allowed to be set to 1 in the patterns [59]. However, in each of the equations given here, it is assumed that the patterns to be stored are completely random.

Hertz *et al.* [77] have shown that the storage capacity of a Hopfield net, in terms of the number of $N$-bit patterns that can be stored, has a theoretical maximum value of $0.138N$ patterns if we are willing to accept a 1% error that any given bit in any training pattern is unstable (*i.e.* that any bit of a training pattern presented as input will not remain the same on iteration), *i.e.*. This gives a critical load value of 0.138. Amit [5] also gives a proof that the storage capacity is proportional to $N$. The exact value depends on the complexity and similarity of the patterns. In general when critical load is calculated for Hopfield nets, the assumption is made that the patterns to be stored are completely random and that a small error in recalling bits is permissible. Increasing the number of stored patterns beyond this limit increases the probability that any particular input pattern will cause the net to iterate to a stable pattern that does not correspond to one of the trained patterns.

The storage capacity of the Hopfield net is generally given in terms of the maximum number of patterns that can be stored, it should be remembered that each of these pattens is $N$ bits long. It is therefore correct to say that the Hopfield can store a maximum of $0.138N^2$ *bits* of information. The critical load for Hopfield nets is quoted as being 0.138 so often that some researchers, such as [20], refer to it as the *Hopfield value.*

However, a few researchers disagree, or claim that they can improve on the figure of $0.138N$ by adapting the architecture a little. Davey *et al.* [38] outline two algorithms that promise high-capacity variations on the Hopfield net. Both involve sacrificing the symmetry of the weight matrix ($t_{ij}$ is not always the same as $t_{ji}$), and it is no longer certain that input patterns result in convergence with asynchronous updating. They define a *local field* for each node $i$, $h_i$, as follows:

$$h_i = \sum_{j \neq i} w_{ij} S_j \tag{2.5}$$

where $S_j$ is the state of the $i$'th node, equivalent to $\mu_i(t)$ in table 2.1 on page 40. For any stored pattern $\xi$ $(\xi_1 \ldots \xi_N)$, the *aligned local field* is given by $h_i \xi_i$. If the

aligned local field for each node is non-negative, the pattern is stable.

The first algorithm given by Davey *et al.* is perceptron-style learning, proposed by Gardner [59]. Gardner claims that applying perceptron-like learning to associative networks of Hopfield-like cells gives a maximum storage capacity of $2N$ patterns if the patterns are uncorrelated, and that this maximum increases if the patterns are correlated. This type of learning is designed to raise the aligned local field for each training pattern above a specified threshold, $T$. The only criterion for the patterns to be learned is that $T > 0$ for all patterns. The second algorithm is *Iterative Local Learning*, proposed by Diederich and Opper [42], in which an iterative algorithm, similar to the perceptron learning rule, is used to ensure that the local fields for each node are either greater than $+T$ or less than $-T$ as appropriate.

They assess the performance of their networks not in terms of the probability of error in any particular retrieved bit, but in terms of the mean radius of the attractor basins for each of the training patterns, $R$, normalised as a proportion of the network size $N$. They quote experiments that show that both training algorithms give $R$ values greater than 0.5, leading to the researchers claiming that all training patterns can be retrieved with high fidelity, although this high performance is achieved at the cost of greatly increased training time, since the training algorithms are now iterative.

MacKay [120] explains that the number of patterns that can in principle be distinguished by a single perceptron is $2n$, where $n$ is the number of input connections to that perceptron. He does not discuss the storage capacity of Hopfield nets as such, but it may be possible to extend his conclusions to cover Hopfield nets. Each cell in a Hopfield net has $N-1$ connections (*i.e.* a connection to every cell other than itself), in which case, its storage capacity is $2(N-1)$. This only differs from the capacity proposed by Davey *et al.* by 2 patterns.

One variation on the simple Hopfield net is the Exponential Correlation Associative Memory (ECAM). Chiueh and Goodman [33] point out that the Hopfield net is simply a specific instance of a Recursive Correlation Associative Memory (RCAM), *i.e.* a network of cells with recursive connections designed to relax to a stable state which it has learned. The general rule for updating $\mu_i$, the state of the $i$th cell, at time $t+1$ is given by equation 2.6

$$\mu_i(t+1) = sgn\{\sum_{s=1}^{M} \xi_i^s f(\sum_{j}^{N} \mu_j(t)\xi_j^s)\} \tag{2.6}$$

where $f$ is a weighting function that is continuous and monotone non-decreasing in the interval $\{-N, N\}$, and $\xi_i^s$ is the state (1,–1) of node $i$. This corresponds to the updating rule (steps 3 and 4 of Table 2.1 on page 40). Replacing $f$ with the sign function gives the behaviour of the Hopfield net. In the case of the ECAM, $f$ is replaced by the exponential function:

$$\mu_i(t+1) = sgn\{\sum_{s=1}^{M} \xi_i^s e^{\sum_j^N \mu_j(t)\xi_j^s}\} \tag{2.7}$$

The architecture was proposed by Chiueh and Goodman [33], who claim that it had an storage capacity that is exponential in the length of the bit patterns (*i.e.* the storage capacity for patterns of length $2n$ is the square of that for patterns of $n$ bits) under certain conditions, specifically when the applied test pattern which is to be classified is close to one of the library patterns in terms of Hamming distance. Hancock and Pelillo [69] claim that the storage capacity of such a network has an upper limit of $2^{N-1}/N^2$, but this claim has been criticised by Wilson and Hancock [189], who claim that the maximum number of storable patterns is in fact:

$$M = 1 + \sqrt{\frac{2\pi[1+4p(1-p)]}{N}} \exp[\frac{N(2p-1)^2}{2+8p(1-p)}] \tag{2.8}$$

where $p$ is the acceptable upper limit on the probability of any particular bit in the retrieved pattern being incorrect.

Bogacz *et al.* [19] claim that the capacity of a Hopfield net can be extended to $0.023N^2$ patterns, provided that one sacrifices the ability to retrieve stored patterns, requiring the net simply to recognise whether a presented pattern is novel or not. To achieve this, they use the energy function of the Hopfield net, a function that gives a measure of how far away from a stable state the net is. The energy function is analogous to the height of a ball rolling around on a smooth landscape: As the Hopfield net relaxes from an unstable initial pattern to a stable one (with luck, one of the stored patterns), the ball rolls around the landscape until it reaches a minimum point, at which it stops. The energy function $E(x)$ for an $N$-bit input pattern $x$

$(x_1 \ldots x_N)$ is defined as

$$E(x) = -\frac{1}{2}\sum_{i=1}^{N} x_i \sum_{j=1}^{N} x_j . w_{ij} \tag{2.9}$$

The energy function is defined to be lower for stored patterns and higher for other (random) patterns. For any given training pattern, $\xi = \xi_1^S \ldots \xi_N^S$, the energy function rearranges to give

$$E(\xi) = -\frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N}(\xi_i^S . \xi_j^S)^2 - \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{s=2}^{P}\xi_i^S . \xi_i^s . \xi_j^S . \xi_j^s \tag{2.10}$$

where $\xi_i^s$ is bit $i$ of the $s$th training pattern ($s = 1 \ldots P$). Since $\xi_i^S$ is either 1 or $-1$, $(\xi_i^S . \xi_j^S)^2 = 1$ and the first term reduces to $-\frac{N}{2}$. The second term is a noise term that can, according to Bogacz *et al.*, be approximated by a normal distribution with mean 0 and standard deviation $\frac{1}{2}\sqrt{2P}$. Similarly, it can be shown that the energy function for a random pattern of bits can be approximated by a normal distribution, also with mean 0 and standard deviation $\frac{1}{2}\sqrt{2P}$ (identical to the energy function for a stored pattern except without the first term). The average energy value for a stored pattern is $-\frac{N}{4}$ and for a random pattern is 0. Bogacz *et al.* take any applied input pattern with an energy value of less than $-\frac{N}{4}$ to be a familiar (stored) pattern and a novel pattern otherwise.

An error occurs if the noise (the Hamming distance between the input pattern and the nearest stored pattern) is higher than the threshold, $\frac{N}{4}$. Taking an acceptable error rate of 1%, then if $P_{max}$ is the maximum number of patterns that can be stored, then:

$$Pr(\theta(0, \frac{1}{2}\sqrt{2P_{max}}) < \frac{N}{4}) \geq 0.99 \tag{2.11}$$

where $\theta(0, \frac{1}{2}\sqrt{2P_{max}})$ is the normal distribution with mean 0 and standard deviation $\frac{1}{2}\sqrt{2P}$. This equation can be solved using the standard normal distribution curve:

$$P_{max} \approx \frac{0.185}{8}N^2 \approx 0.023N^2 \tag{2.12}$$

Graham and Willshaw [64] have produced a measure termed "information efficiency", defined as the ratio of the amount of information that can be retrieved from the memory to the amount of storage available. They quote information efficiencies of up to 69% for their heteroassociative memories for sparse patterns. The network that they describe is sufficiently different to both cell assemblies and Hopfield nets to make any direct comparison pointless. However, the concept of information efficiency itself is one that can in principle be applied to both these types of network.

This section has shown that various researchers have produced different formulae for storage capacity. Although the researchers all describe their formulae as referring to associative memories, this is a general catch-all term, and, in practice, all the architectures differ to a greater or lesser degree. Furthermore, different assumptions are made, such as the probability of an acceptable incorrect bit in a retrieved pattern and the exact nature of the patterns recalled (such as the novelty detection architecture proposed by Bogacz *et al.*). Each equation assumes that the patterns stored are completely random, which, in a practical system, is never the case. There is reason to believe that the capacity of an associative net increases as the proportion of 1 bits in the patterns stored decreases.

The one thing that can be concluded from all this is there appears to be no definitive formula for the capacity of attractor nets. Specifically, I am not aware of any previous research that has been done on the storage capacity of networks of cell assemblies, and that was the motivation for the research described here.

## 2.4 Genetic Algorithms

Experiments in subsequent chapters rely upon a genetic algorithm [82] to determine parameter values, so a brief introduction to genetic algorithms is included. A genetic algorithm is a method of searching a vast search space in situations where an exhaustive search is impractical or would take too long. As the name implies, it is inspired by genetics and evolution and embodies the principal of survival of the fittest to determine a suitable, although not necessarily optimal, solution to a problem.

Potential solutions take the form of chromosomes, which are patterns of num-

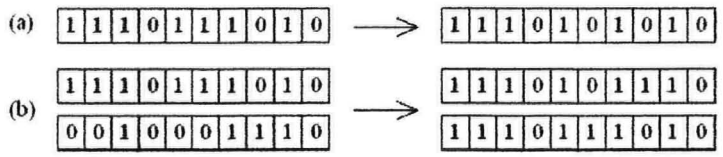


Figure 2.9: Mutation and cross-over are the basic mechanisms of evolution in genetic algorithms. In (a) the sixth bit is mutated from 1 to 0. In (b) a cross-over occurs in a pair of chromosomes with a split occuring between the fifth and sixth bits and the latter halves of each chromosome swapped.

bers that encode the behaviour of some model. A chromosome representing some game-playing strategy, for example, might encode that strategy in the form of a few simple rules. Genetic algorithms usually implement chromosomes as binary patterns, although this need not be the case. Each chromosome is translated into a phenotype, that demonstrates the behaviour encoded in the chromosome.

A population of random chromosomes is created and each corresponding phenotype evaluated according to a suitability metric. Only the chromosomes that score the highest are retained to the next generation. The next generation consists of mutations of the surviving chromosomes or crosses-over, as shown in figure 2.9. In some cases, unchanged copies of the surviving chromosomes are also retained. Mutations consist of copies of chromosomes in which one element has been changed randomly, either within a small range of its original value or within the entire possible range of that element. In the case of binary elements, a mutation involves inverting one bit randomly. A cross-over is akin to sexual reproduction in nature, in which two chromosomes are split at the same random point along their length and the two end sections swapped. Further swapping of sections can take place at more than one point along chromosomes. Typically cross-over results in faster evolution as it allows advantageous sections of chromosomes to be combined.

As evolution progresses, the average fitness of each generation of chromosomes is greater than that of the previous one, until a maximum value is achieved. It is possible that this maximum is a local one in the entire search space, and to avoid this, many genetic algorithms allow a small number of purely random chromosomes

to be added to each generation. These allow a population an escape from a local maximum although many generations may be needed before the escape happens.

The above description outlines only the basic format of a genetic algorithm. Other optimizations are possible, but for the purposes of the experiments described in this thesis, it was found that the simple strategy presented here was sufficient to produce acceptable results.

For genetic algorithms to be effective, the following criteria must be fulfilled:

1. The search space must be so large as to make an exhaustive search impractical. Although a genetic algorithm is guaranteed to find the optimal solution if left to run for an infinite length of time, in practice there is no way of knowing whether the final solution is the best possible. For a small search space, it makes more sense to carry out an exhaustive search.

2. It must be possible to assign a score to each chromosome that indicates its degree of fitness. Categorising a chromosome simply in terms of success or failure would not be enough.

3. The fitness scores of adjacent points in the search space should usually be similar, so that a gradual improvement in fitness can be achieved by tracing a path through that space.

The description above outlines the general principles behind genetic algorithms. Variations are possible, of course. For instance, the fitness function may be changed after a certain number of generations in order to fine-tune the evolution. In general, genetic algorithms represent a powerful method of finding a satisfactory solution within a reasonable time in situations where more systematic searching is impractical.

Experiments described in later chapters use a genetic algorithm to assess the performance of networks of cell assemblies under different circumstances. The configurations of the networks are determined by a small set of parameters, the values of which form the chromosomes. Each chromosome is translated 10,000 times to produce 10,000 networks with random connection destinations but the same underlying topology. Each network is then run for 300 time steps under the appropriate external

activation conditions. For each time step a count is maintained of the number of cells that fire in assemblies that are supposed to have ignited and the number that fire in cells that are not. The total score for each chromosome is the total sum of the desired firings less the total sum of the undesired ones. In this way, parameter sets are determined that represent the best compromise between two competing tendencies, namely, desired and undesired ignition of cell assemblies.

## 2.5 Summary

This chapter has described briefly some of the areas of research that have contributed to the work in this thesis. Some of the topics that are covered do not appear at first sight to be linked in any way, but they all form strands that will be woven together in later sections. The chapter has inevitably concentrated on the development of what little theory exists about Cell Assemblies and the principle of Hebbian learning which lies behind it. Although cell assemblies were originally conceived as a description of neural structures in the brain, they have been adopted by connectionists as a neural network architecture. Hebb himself, presumably, never conceived of cell assemblies in terms of computer simulation!

Although several researchers have implemented cell assemblies in computer programs there is little theory describing their behaviour. For this reason, their close cousins in the connectionist family tree, Hopfield nets, whose properties are well known, are also described in the chapter. Experiments in subsequent chapters do show that cell assemblies and Hopfield nets do behave similarly in some ways, but there are also important differences. Hopfield nets always store patterns with the same number of bits as there are cells in the net. The patterns stored by cell assemblies do not store patterns on a bit-by-bit basis. Instead groups of cells are activated together in each cell assembly. Whereas Hopfield nets generally reach stable states in which no cell output changes from one time step to the next (although some patterns do result in oscillation), cell assemblies depend on cell activity dying out and then being rekindled from other cells in the cell assembly. In this way, cell assemblies achieve pseudo-stable states rather than the rigid stable states of Hopfield nets. This

does give them one advantage over Hopfield nets, insofar as they can move from one pseudo-stable state to another. This may be the basis of thought progression and sequential reasoning in the brain, as Hebb proposed [74, 76].

The chapter included a brief description of genetic algorithms, since a genetic algorithm was used to estimate the parameters in experiments described in later chapters. Genetic algorithms borrow the principles of evolution and survival of the fittest to produce a satisfactory solution to any problem in which an exhaustive search of all possible solutions is impractical. Simulated cell assemblies depend on the settings of a small number of parameters from an infinite parameter space. The possible solutions are encoded in the form of numerical patterns known as chromosomes. For genetic algorithms to be effective, it must be possible to translate the chromosomes into scores, which can then be rated, and it will be seen in subsequent chapters that the parameters controlling the behaviour of the cell assemblies in the experiments fit this description perfectly. Although genetic algorithms cannot in practice be guaranteed to achieve the best possible results, they are a powerful tool for adequate parameter values in situations where other means may not be practical.

Chapter 3 on page 53 puts the general concepts outlined in this chapter into a more rigid mathematical context. It also defines in detail the model that will be used for all the experiments described in this thesis.

# Chapter 3

# Description of the Network Model

This chapter describes work carried out to implement a simple hierarchy of cell assemblies in a network of cells as a computer simulation. It is shown that cell assemblies can be constructed by setting the strengths of the connections between cells to specific values, and that cell assemblies can develop naturally as a result of weights being learned in response to repeatedly presented input patterns. The complexity of these cell assemblies is gradually increased in subsequent experiments, and different combinations of cell assemblies are investigated as to their feasibility. Experiments are described that demonstrate that the principle of learning weights can be used to create a hierarchy of cell assemblies.

The chapter starts with a description of the simulation on which all experiments were carried out. This implements a network of generic cells exhibiting the basic concepts of activation, fatigue, recovery and retention. Although previous simulations have attempted to model specific types of neuron with great accuracy [96], it was felt that this was not necessary to produce useful results, so only these basic functions found in neurons, were implemented (see Chapter 2 on page 31).

The behaviour of the model depends on the values of nine global parameters, each of which can theoretically take an infinite number of values. Since it is clearly impossible to test the parameter space exhaustively, a strategy was needed to find an acceptable set of parameter values. A simple genetic algorithm (see section 2.4 on page 48) proved sufficient in this regard.

The experiments were carried out as a series of programs on a standard personal

computer using a mixture of Java and C++. Each simulation was based on a series of discrete time steps, and each cell in the simulation was considered to produce at most one output per time step. This is equivalent to an activity spike produced by a real neuron. During a time step a cell can either fire or refrain from firing. It is therefore an easy matter to translate the firing pattern from a simulated cell to the sort of trace shown in figure 2.6, with each time step being equivalent to approximately 10ms.

Cell assemblies in the simulation are termed either *primitive* or *compound*. The programs simulate a number of cells, each of which is associated with a certain primitive cell assembly. A primitive cell assembly is one that is self-contained and invariant. It contains no cells that form part of any other primitive cell assemblies. A compound cell assembly consists of a grouping of primitive cell assemblies. Primitive cell assemblies are typically referred to by letters: $A$, $B$, $C$ etc. and compound cell assemblies by strings of letters, e.g. $ABC$ refers to a compound cell assembly consisting of primitive cell assemblies $A$, $B$ and $C$. Each cell assembly in the simulation, primitive or compound, was developed according to principles outlined below to obey Sakurai's [154] five defining points (see section 1.1 on page 4). The concept of representing individual primitive cell assemblies by letters of the alphabet and cell assemblies corresponding to combinations of concepts by strings of letters has been proposed by researchers such as [20].

## 3.1 Mathematical Description of Cells

The structure and function of each cell in the simulation was chosen to implement some simple functions of brain cells, rather than any specific type of neuron. The cells implemented are described as fatiguing spiking leaky integrators [91, 140]. Each cell was defined by the following features:

*Activity.* The cell possesses a level of activity, defined as a floating point number limited to the range 0 to 1. This represents the amount of "energy" that the cell has acquired through its connections and/or external activation and is analogous to the membrane potential of real neurons.

*Fatigue.* Each cell possesses a fatigue level, defined as a floating point number

that indicates how tired the cell has become. Whenever the cell fires, its fatigue level increases by a certain fixed amount, termed the *fatigue rate*. Whenever the cell does not fire, its fatigue level decreases by a certain fixed amount, the *recovery rate*, that is not necessarily the same as the fatigue rate. The fatigue level of the cell is constrained to the range 0 to 1. Fatigue imitates the property of biological neurons that prevents them firing indefinitely in rapid succession. This effect causes our skin to lose sensitivity to our clothes, with the result that we are not distracted by them all the time, and to ignore the odour of cigarette smoke in a room after a short period.

*A firing threshold.* This is a floating point number in the range 0 to 1 common to all cells which determines whether they fire or not. Firing is the process of transmitting a signal to other cells in the network, and is determined by the activity of the cell and its fatigue. If the activity of the cell minus its fatigue is greater than the firing threshold, then the cell fires, and produces a signal, equal in strength to each synaptic weight strength, that is propagated to other cells to which it is connected. Biological neurons produce spikes that are the same amplitude, so the simulated neurons may be thought of as producing an output signal of 1 unit, subsequently modified by the synaptic weight strengths. When a cell fires, its activity level drops to zero.

*Inhibitory or excitatory activation.* Cells are classified as either excitatory or inhibitory. This is in accordance with Dale's Principle [45, 181] which states that a neuron produces and releases only one type of neurotransmitter, effectively limiting it to either excitatory or inhibitory behaviour. This is sometimes erroneously referred to as Dale's Law. There is now evidence that neurons can contain and release more than one kind of neurotransmitter [10], but I have nevertheless adhered to Dale's Principle. When excitatory cells fire, the activation that they provide increases the activity level of destination cells. Inhibitory cells reduce the activation of destination cells when they fire. This behaviour is roughly analogous to the behaviour of excitatory pyramidal cells and inhibitory chandelier cells in the brain. The general consensus of opinion is that excitatory cells are roughly four times as common in the brain as inhibitory ones, so 80% of the cells in the simulation were excitatory [27, 45, 61, 99]. This ratio has been used successfully in other simulations [78, 95].

*Inter-cell connections.* Cells pass activity to each other through a series of weighted

connections, with each cell possessing the same number of connections to others. Each connection has a strength, or weight, limited to the range 0 to 1 (for excitatory cells) or -1 to 0 (for inhibitory cells), indicating the proportion of the signal strength that is passed along the connection.

*Retention.* Incoming activity to a cell is added to the cell's current activity level (or subtracted if it originated from an inhibitory cell). However, cells take the form of leaky accumulators, in so far as only a percentage of the activity is retained, according to a fixed retention rate. This effect is often referred to in the literature as *decay* (examples include [70] and [34]). In biological neurons this decay is caused by the leakage of currents through the cell membrane (see [3]).

The activity of the cell at time $t$ is therefore determined by equation 3.1.

$$a_t = \sum_i w_i + r.a_{t-1} \tag{3.1}$$

where $r$ is the retention rate, $a_t$ is the activity at time $t$ and $w_i$ are the strengths of all the connections to that cell summed over all cells that fire. This is a version of the membrane potential equation for the Integrate-and-Fire Neuron [96] shown in equation 3.2.

$$v^{t+1} = \begin{cases} I - a + bv^t & \text{if } v^t < \text{ threshold} \\ c & \text{if } v^t \geq \text{ threshold} \end{cases} \tag{3.2}$$

where $v^t$ represents the membrane potential at time $t$, $I$ the input current and $a$, $b$, $c$ and threshold are the parameters. When the membrane potential reaches the threshold value, the neuron is said to fire a spike and $v$ is reset to $c$. In the current model, $I - a$ is the summed weighted input, $b$ is the retention rate, and $c$ is 0, the activity level of a cell after it has fired.

Propagation of activity between cells took place simultaneously once per time step. Clearly it is necessary to give some external activity to the network of cells, otherwise activity could not be propagated between cells[1]. During the first $T$ time

[1]Some experiments were carried out in which a small proportion of cells activated spontaneously, but the majority of experiments relied only on external activation to ignite cell assemblies.

steps, cells in primitives that were activated externally had a 40% chance of their activity being boosted to the maximum value of 1. $T$ was held constant throughout any experiment at 10. This extended activation takes place simply in order to allow primitive cell assemblies to build up sufficient energy to maintain activity when the external stimulation is removed. Different primitive cell assemblies are activated in different experiments.

The parameter values shown in table 4.1 on page 77 were chosen as a compromise. Experiments showed that if the proportion of cells being activated was substantially lower than 40% or was applied for fewer than about 10 time steps, then activity in the cell assembly died out quickly after external activation was removed. If, however, the proportion was much higher than 40% or for much longer than 10 steps, a large proportion of the cells fatigued at approximately the same time step, thereby rendering them unable to fire and reducing the activity in the cell assembly to the point where it could not be sustained.

For each cell $i$, the cell fires if the activity of the cell, $a_i$, minus the fatigue of the cell, $f_i$, is greater than the firing threshold, $\theta$. If the cell fires, then for all cells $j$ to which there is a connection from cell $i$, the temporary activity $temp_i$ is increased by the weight strength of the connection between the cells, $w_{ij}$:

$$temp_i = \sum_j w_{ij}$$

The summation is performed over all cells $j$ which provide excitatory or inhibitory activity to cell $i$.

If the cell fires:

1. the activity $a_i$ is reduced to zero.

2. the fatigue level, $f_i$, of the cell is increased by a certain fixed constant, termed the *fatigue rate*, $F$. The fatigue is limited to a maximum of 1.

If the cell does not fire the fatigue level of the cell is reduced by a certain fixed constant, the *recovery rate*. The fatigue is limited to a minimum of 0. Inhibitory neurons function in the same manner except all the weights from them are negative. The temporary activity values of the cells are not limited to the range 0 to 1.

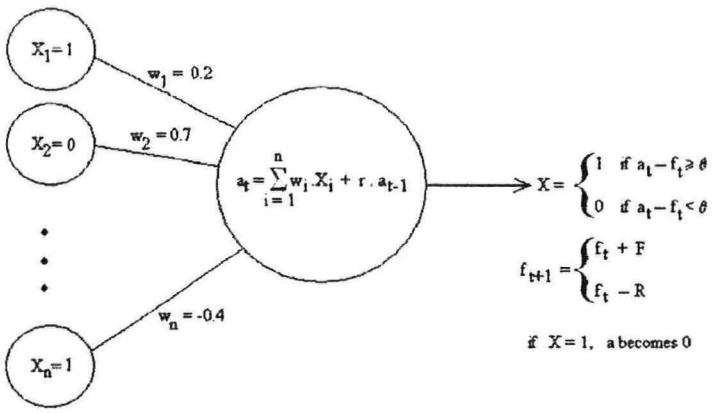


Figure 3.1: A summary of the behaviour of a cell. Circles on the left indicate cells that fire ($X = 1$) or not ($X = 0$) and contribute activity via weighted connections. $a_t$ is the activity level for the current time step, $f_t$ the fatigue level, and $F$ and $R$, the fatigue rate and recovery rates.

The new activity for each cell is determined from its temporary activity. The activity of each cell $i$ is given by a proportion of the activity of the cell, given by the retention rate, at the previous time step (or 0 if the cell has just fired) added to the total temporary activity that the cell has received on this time step. Effectively this implements equation 3.1 on page 56. The activity of the cell is then limited to the range 0 to 1. The upper limit is not really necessary, since the only action that a cell can take is to fire, in which case the activity sent to destination cells is equal to the connection strengths of the synapses. However, it did help to prevent numerical overflow in the computer programs. The lower limit of 0 prevents the unexpected effects that would occur from a cell having "negative energy".

The relationship between the parameters is summarised in figure 3.1. $X$ is a binary flag (1 or 0) introduced for convenience to indicate whether a cell fires or not.

## 3.2 Hebbian learning rule

The majority of experiments described in later chapters investigate the properties of networks of cells in which connection strengths are predetermined. This was done in order to prevent one possible source of variation, individual connection strength, from

affecting the results. In several cases, it was impractical to implement learning, due to either hardware limitations or pressure of time. However, adaptation of connections is one of the essential tenets of Hebb's theory [74], and so several experiments were included in which learning took place. It is therefore necessary to describe the exact format of the learning rule used.

The connections between cells increase when both the source cell and destination cell fire at the same time step. This applies to both excitatory and inhibitory cells alike, although a slightly different learning rule is applied to each type in order to avoid weights going towards zero when they should be becoming more pronounced.

Whenever the source cell is excitatory, the increase is proportional to the difference between the current value of the weight and the maximum value of 1:

$$\text{new } w_{ij} = \text{ old } w_{ij} + \eta(1 - \text{ old } w_{ij}) \tag{3.3}$$

where new $w_{ij}$ and old $w_{ij}$ are the new and current weight values of the connection between cell $i$ and cell $j$ respectively and $\eta$ is a learning rate. It was found that a learning rate of $\eta = 0.025$ ensured that weight changes are fast enough to ensure that the weights approach their desired values within a reasonable time, yet slow enough to avoid wild oscillations around those values. The second term in this equation ensures that learning is stable, and that weights cannot exceed 1.

Whenever the source cell is inhibitory, both cells firing concurrently should cause the weight to rise towards zero. This ensures that the source cell will be less likely to shut down the destination cell if it fires in future. In this case, a suitable formula is as follows:

$$\text{new } w_{ij} = \text{ old } w_{ij} - \eta \text{ old } w_{ij} \tag{3.4}$$

Since $w_{ij}$ is negative, this subtraction causes the weight to rise towards zero.

Hetherington and Shapiro [78] have suggested that cell assemblies can be learned if post-not-pre LTD is used, but not if pre-not-post LTD is used (see section 1.1 on page 4), although other researchers (such as [91]) have achieved good results with pre-not-post LTD. Preliminary experiments suggested that post-not-pre LTD gave

better performance in the case of this particular simulation. A simple post-not-pre LTD rule was therefore implemented as follows.

Whenever the source cell is excitatory, the decrease is proportional to the current value of the weight. This took the form of the same equation as for the LTP of inhibitory cells, except that the weight is decreased as it is a positive figure:

$$\text{new } w_{ij} = \text{ old } w_{ij} - \eta \text{ old } w_{ij} \tag{3.5}$$

Whenever the source cell is inhibitory, LTD should encourage the inhibitory cell to shut down the destination cell whenever it fires, *i.e.* it should decrease the negative weight away from zero. This may be achieved by the following formula:

$$\text{new } w_{ij} = \text{ old } w_{ij} - \eta(1 + \text{ old } w_{ij}) \tag{3.6}$$

These rules were chosen to ensure that weight changes were reduced as the weights approached extreme values. For example, as an excitatory weight increased, further increases became smaller. Clearly, the final value of weights between any given pair of cells will depend on the frequency with which they fire together. Using the rules described above implies the following two results: The more often cells fire together, the closer to 1 weights from an excitatory cell will be, and the closer to 0 weights from an inhibitory cell will be. The less often cells fire together, the closer to 0 weights from an excitatory cell will be, and the closer to -1 weights from an inhibitory cell will be. In general, the weights adapt to reflect the proportion of the time that the post-synaptic cell fires given that the pre-synaptic cell has fired.

The size of the learning rate affects the stability of the learning process. Figure 3.2 shows the result of mathematical simulations in which a weight between two cells was adjusted in isolation and the frequency of co-occurrence of activation was maintained at 40% of time steps. The initial values of the both excitatory and inhibitgory weights were set to 0, as equation 3.3 allows a zero weight to increase rapidly, and equation 3.6 allows inhibitory weights to drop away rapidly from 0. These simulations suggest that after a large number of training steps, excitatory weights will approximately match the proportion of time steps for which the two cells fired together, and inhibitory weights will be the negative equivalent of the proportion of time steps for which they

did not fire together. This tendency can be proven mathematically. For example, if cells on either side of a connection fired together 80% of the time, then an excitatory weight between them would be 0.8 and an inhibitory weight would be -0.2. I term this the *negative equivalent weight.* Increasing the learning rate means that the weight reaches the predicted proportions rapidly, but then fluctuates wildly. Decreasing the learning rate gives more stable behaviour, but the weight now takes much longer to reach its final value.

However, this simple analysis does not take into account the possibility of reverberation of the cells, not present in the mathematical simulations shown in Figure 3.2, in which during the training phase, cells are stimulated to fire as a result of activity passed to them by other cells rather than as a result of external activation. This effect would become more noticeable as training progressed as connection strengths increased between externally stimulated cells, and would produce the same effect as a higher co-firing rate between the cells.

## 3.3 Network topology

This section describes how a network may be constructed from cells described in section 3.1 on page 54. In each simulation 150 cells were associated with each primitive cell assembly.

The figure of 150 was chosen as a compromise. The more cells that are present in a cell assembly, the easier it is to maintain activity in that cell assembly when it is activated. Also, a large number of cells facilitates the learning of cell assemblies as the number of cells concurrently active during training increases. However, a large number of cells does increase the running time of any simulation, and the programming environments used imposed their own memory limitations. Early experiments were carried out using networks of 400 cells. These experiments were repeated with networks of 150 cells with almost identical results.

Cells were implemented as a one-dimensional array. Connections between cells were assigned at random as shown in figure 3.3, other than cells were prevented from having any connection to themselves, as no cell in the brain has a self-connection. The

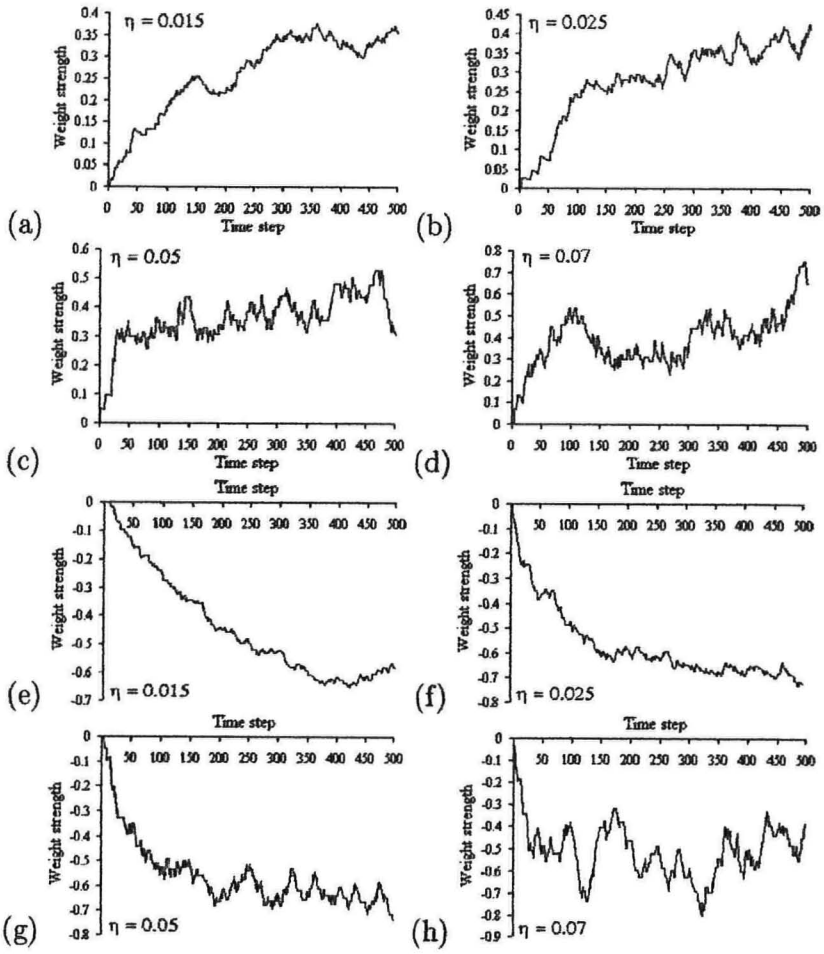


Figure 3.2: Simulations of the adjustment of a weight in isolation for several learning rates ($\eta$) as indicated for an excitatory connection (a-d) with an initial weight value of 0, and an inhibitory one (e-h) with a very small initial value, 0.001. Small learning rates do result in the weight approaching its predicted value reasonably quickly. The graphs shown in this figure represent a typical run.

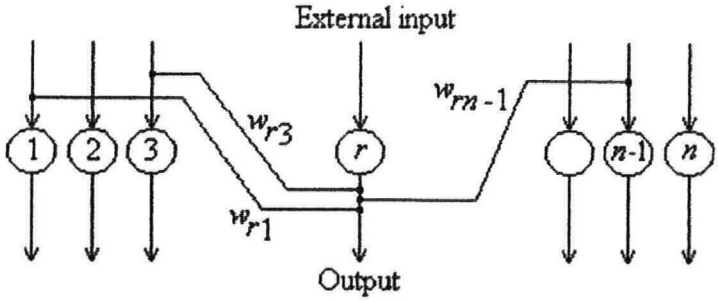


Figure 3.3: Connections between cells in the network

number of connections emanating from each cell varied according to the particular experiment being carried out.

The exact behaviour of the network is determined by ten fundamental parameters. The first six are predetermined constants for any network insofar as they are not subject to Hebbian learning even in experiments in which Hebbian learning was implemented.

1. The number of connections per cell - the number of cells to which any given cell in the network provides output.

2. The fatigue rate - the value by which the fatigue level of any cell increases every time that it fires.

3. The recovery rate - the value by which the fatigue level of any cell decreases on every time step that it does not fire.

4. The firing threshold - the activity level above which cell activity causes firing, plus fatigue.

5. The retention rate - the proportion of the activity of any non-firing cell that is retained from one time step to another.

6. The probability of any particular cell being excitatory.

7. The weights from excitatory cells to other cells within the same primitive cell assembly.

8. The weights from inhibitory cells to other cells within the same primitive cell assembly.

9. The weights from excitatory cells to other cells that are within a related primitive (*i.e.* within a compound assembly).

10. The weights from inhibitory cells to other cells that are within a related primitive.

The last four parameters are weights between individual cells. These may be predetermined in cases where cell assemblies are created artificially, or they may evolve as a result of learning and repeated presentation of input patterns. Predetermined weights are determined simply by the identities of the primitives in which the pre-synaptic and post-synaptic cells reside. For instance, if two primitives are strongly related within a compound cell assembly, excitatory weights between them will be relatively high and will all have the same strength. This approach has the advantage that individual weight strengths need not be stored as separate entities, but may be calculated as and when they are needed. Storage of individual weight values is not a problem in smaller networks, but the memory requirements become prohibitive when large networks (typically more than about 1000 primitives) are implemented.

Weights between cells within primitives that do not form part of a compound assembly are not listed as parameters since they should never fire at the same time. Excitatory weights between such cells should therefore be 0, and inhibitory weights set to an extreme negative value in order to discourage co-firing.

The simple learning algorithm explained in section 3.2 on page 58 predicts that the weights from inhibitory cells should be strongly correlated to those from excitatory ones. In order to simplify and accelerate the evolutionary process, in early experiments, the inhibitory weights were automatically derived from excitatory ones, and it was found that networks based on such weights still gave good performance. The automatic correlation between inhibitory and excitatory weights was lifted for the large scale experiments described in chapter 5 on page 121 in order to optimise performance.

## 3.4 Parameter estimation using a genetic algorithm

A large part of the work involves determining parameter values that promote optimal (or near optimal) performance. An exact mathematical analysis of the problem is beyond the scope of this thesis as any simulation based on this system contains a large stochastic element. A lack of rigorous mathematical analysis precludes the possibility of determining optimal values for the parameters. For this reason, it was decided to use a genetic algorithm to determine acceptable values (see section 2.4 on page 48). The genetic algorithm was based upon chromosomes of nine values, consisting of all of the values listed directly above with the exception of the probability of a cell being excitatory, which was kept at 80%. This value was chosen to mimic the proportion of excitatory pyramidal cells believed to exist in the cerebral cortex [68]. All the parameters were floating point numbers restricted to the range 0 ... 1, except the number of connections per cell, an integer limited to the range 1 to 50 (chosen so as not to exceed memory capacity of the computer), and the weights from inhibitory cells which were necessarily negative, limited to the range -1 ... 0.

A population of 100 chromosomes with random values was created. The size of the population was limited by computational tractability. Each was translated 10,000 times independently into networks of cells and these networks were assessed according to certain criteria. In Experiment 4.1.1 on page 75, each network was run for 300 time steps and the total number of cells firing at each time step was recorded. The score for each chromosome in this case was the average of these totals over each run, so the genetic algorithm in this case favoured networks containing cell assemblies that ignited quickly and persisted for a long time. In subsequent experiments, in which combinations of primitives were used, more complex scoring methods had to be used. For instance, in experiment 4.2 on page 80, three primitives were to be linked in such a way that activating any two primitives was sufficient to ignite the third, and yet activating one single primitive was insufficient to ignite either of the other two. It would be an easy matter to find parameter sets to accomplish either of these tasks without the other: Large numbers of connections together with strong excitatory weights between cells would easily allow two primitives to ignite the third, for instance. In this situation, the 10,000 runs were divided evenly into runs in which

two primitives chosen at random were activated and runs in which one primitive was activated. The scoring method was to subtract the total number of undesired cells that fired at any time step on the second type of run from the total number of cells that fired at any time step on the first type of run. Similar scoring methods were adopted for more complicated networks (chapter 5 on page 121).

When all the scores for each generation had been determined a simple function was used to breed the next. The highest scoring ten population members were retained unaltered. Forty population members were created by mutating one gene (parameter) from each of the ten retained ones. A mutation comprised the replacement of the parameter by a random value within the legal range for that parameter. Forty population members were created by crossing two of the ten retained members. The crossing entailed choosing each parameter value in turn at random from the two "parent" chromosomes. The final ten members of the new population were created from totally random values within the legal range for each parameter. The random chromosomes were chosen in order to avoid the system getting stuck at a local maximum in the parameter space.

Repeated use of the simple genetic algorithm from completely random starting points produced an interesting range of results, with the local minima in the parameter space being achieved within 30 generations in each case. The success rates for each of the trials were similar, and due to stochastic variations in the test runs themselves, it was impossible to choose a definitive "winner". The defaults for the six predetermined parameters shown in Table 3.1 produced good performance, and as a consequence, they were chosen for experiments 4.1.1 and 4.1.2 (on pages 75 and 79 respectively).

A similar approach was used to create the behaviour of the simulated cell assembly shown in figure 2.5, which emulates the behaviour of the TRACE model [98]. In this case, the score was derived by dividing the number of cells that fire at any given time step by the total number of cells and then calculating the square of the difference between this value and the equivalent value produced by the TRACE simulation for the same time step. Chromosomes with lower summed square differences are preferred.

| | |
|---|---|
| Number of connections per cell | 20 |
| Fatigue rate | 0.19 |
| Recovery rate | 0.09 |
| Firing threshold | 0.95 |
| Retention rate | 0.8 |
| Probability of excitatory cell | 0.8 |
| Weight strength between cells in the same primitive | 0.44 (-0.56) |
| Weight strength between cells in different primitives | 0.08 (-0.92) |

Table 3.1: Default values of parameters. Only the weight strengths were derived from Hebbian learning. The inhibitory weight strengths (in parentheses) were derived from the excitatory weight strengths rather than being evolved independently

It was necessary to change the number of connections per cell as the size of the network increased in order to ensure that the average number of connections between cells within and between primitives remained roughly constant. This is necessary to ensure that the primitives can pass enough activation between each other to maintain activation. The level of activation from one primitive to another depends on the number of connections and the weight strengths between primitives. If the weight strengths are kept the same as the size of the network increases, then the number of connections must be increased to maintain the same effect. The default value of 20 for the number of connections was determined for three primitive cell assemblies in a simple *ABC* network. Increasing the number of connections per cell can be justified biologically speaking by the fact that it has been estimated that the average number of connections from each cell in the rat hippocampus is approximately 2000. Such a number would clearly be impractical in a simulation due to pressures of time and memory space given the limited resources at my disposal.

The fact that the genetic algorithm was run several times from independent starting points allows general patterns to be sought. Surprisingly, some parameters that one would expect to be strongly correlated turn out to have a weak correlation at best, such as that between the number of connections and the excitatory weight strength for connections within a primitive ($R^2 = 0.0005$). Appendix B on page 190

explains how the significance level may be derived from the coefficient of determination. Parameters that do have some correlation ($R^2 > 0.1$, approximately 20% significance level) do not have an immediately obvious relationship, and this only goes to illustrate the complex interaction between the nine different parameters and the difficulty in producing a mathematical analysis of the system. Figure 3.4 shows scattergrams relating to the parameters for experiment 5.5 on page 139. Comparisons of parameters that yielded a coefficient of determination ($R^2$) of less than 0.1 have not been included. In general, the small number of runs of the genetic algorithm (12) means that only a handful of the coefficients of determination can be said with any certainty to be significant at the 20% level or better.

The strongest correlation occurs between the firing threshold and the excitatory weight strength between cells in different primitives, with a coefficient of determination of 51% (figure 3.4(a)), significant at the 1% level. One would generally expect a positive correlation between these due to the tight constraints on the energy passed between primitives for correct behaviour. In the 3-4 networks implemented in experiment 5.5 on page 139 a primitive should ignite when receiving energy from three designated primitives, but not when receiving energy from only two of those three. This places an upper and lower limit on the incoming activity energy required to ignite the primitive. As the excitatory weight strength for inter-primitive connections increases, the firing threshold must increase similarly to accommodate the increased energy entering the primitive. If it does not, then the number of cells firing in the primitive increases, and this can cause runaway activity.

These graphs should not be confused with those shown in figure 4.6 on page 85. Those show how the variation of two parameters affects success rate while the others remain constant, whereas the graphs shown in figure 3.4 represent different sets of parameters. Nevertheless, the graph in figure 4.6(c) does show a similarity to figure 3.4(f).

This explanation also accounts for the correlation between the number of connections per cell and the excitatory weight strength of inter-primitive connections (figure 3.4(f)). The energy is a function of the inter-primitive weight strengths, both excitatory and inhibitory, and the number of connections between cells (a reflection of the

total number of connections between primitives). Table 5.2 on page 140 shows that the genetic algorithm settled on inhibitory weights close to zero, so the total energy is approximately proportional to the product of the excitatory weight strength and the number of connections. Given the tight constraints on energy, as the number of connections increases, the weight strength should decrease. One would therefore expect a negative correlation and this does turn out to be the case. A similar argument can be made for the fairly strong negative correlation between the number of connections per cell and the retention rate. The higher the retention rate, the easier it is for a cell to achieve the activation level necessary to fire and the easier it is for a primitive to ignite.

Figure 3.4 suggests that one parameter value is more critical than the others. Although the genetic algorithm was repeated several times, the excitatory weight strength between cells always evolved to a small range of values, centred around approximately 0.08. The other parameters evolved a wider range of values. Interestingly, the excitatory weight between cells in different primitives has relatively little correlation with the number of connections between cells. Equation 4.1 on page 97 shows that the total activation "energy", loosely defined as the total excitatory signal strength, transferred between primitives is approximately proportional to the product of these two parameter values. One would expect that increasing the number of connections per cell while keeping the excitatory weight strength between primitives roughly constant would increase this activation energy and lead to runaway activation. However, a statistical analysis of the activation energy arriving at each cell showed that the primitives were ignited by relatively few cells within them, cells which through stochastic variation happened to be well connected. I term these cells "lucky neurons". The destinations of connections are assigned randomly throughout the network with the result that some cells have more incoming connections than others. These cells are therefore activated more readily as they generally receive more incoming energy than less well connected cells. Activating only a few of these lucky neurons (typically between 10 and 15) within a primitive is enough to ignite it. A trial of 10,000 runs shows that they occur in most runs (more than 95%) and that their presence spoils any relation that one might expect to exist between

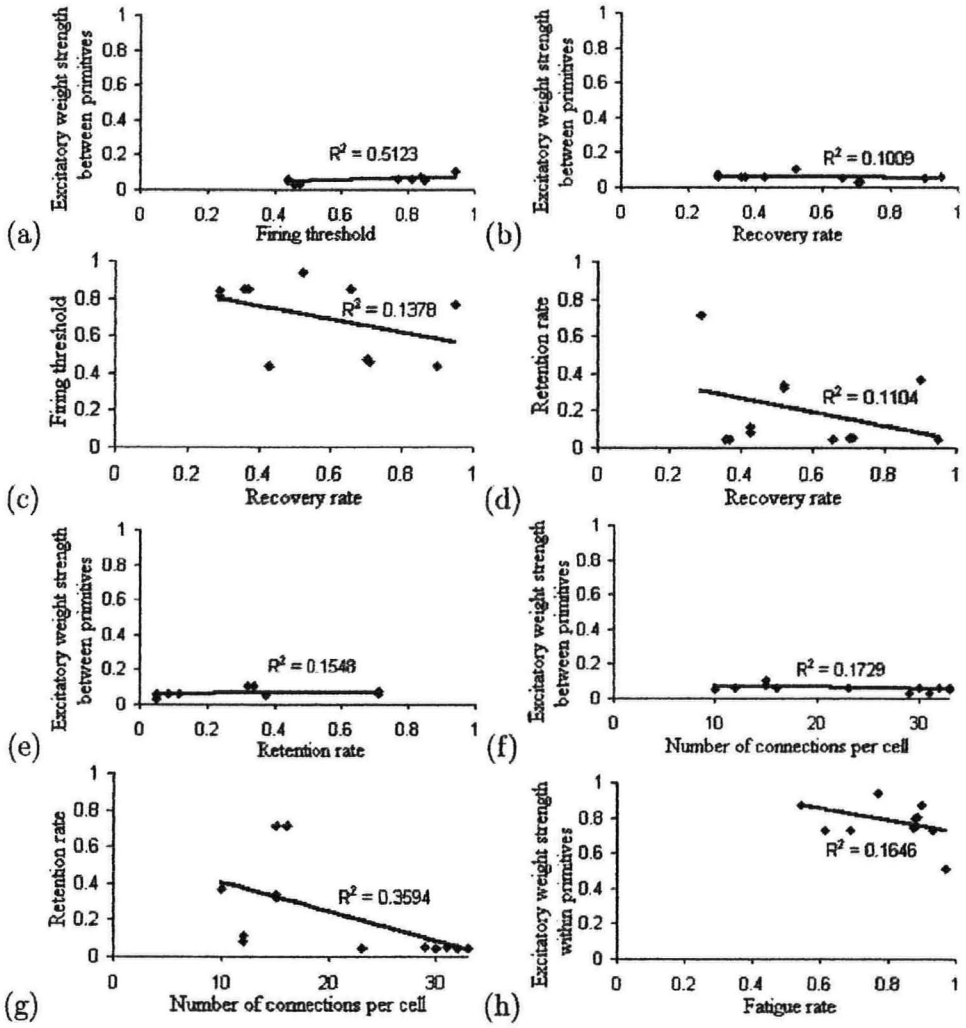


Figure 3.4: Scattergrams illustrating correlations between various parameters estimated for one cell assembly network. The significance levels of the coefficients of determination are listed in table B.2 on page 192.

number of connections per cell and the excitatory weight strength between cells in different primitives. The presence of connections from cells to themselves would tend to increase this problem, but care was taken not to implement such connections.

## 3.5 Summary

This chapter formalises some of the concepts described in chapter 2 on page 31 in more mathematical terms. Hebb did not formulate his theory as a set of equations, which gives a degree of flexibility. Section 3.2 on page 58 gives a set of equations both for updating weight strengths to take account of co-firing of cells and to propagate activity from one cell to another. These give adequate performance, as will be shown by later experiments. The equations predict that weight strength will correspond to the percentage of occasions on which both the presynaptic and postsynaptic cells fire, and a simulation shows that this does generally occur more or less.

The last section of the chapter outlines the essential features necessary to get a simple cell assembly working. These can be reduced to nine parameters values which co-operate to give the required behaviour. Determining the optimum parameter values is beyond my capabilities and the parameter space is too large to be searched exhaustively, and the complex interaction of parameters defeated my meagre mathematical attempts to calculate the optimum values. However, the nature of the problem readily lends itself to solution by a genetic algorithm, and repeated use of such an algorithm produces several solutions all essentially as good as each other. One such evolved parameter set, that for a 2-3 network has been given for illustration purposes. Networks of primitives in subsequent chapters become more complex and different parameter values are required, but the same basic genetic algorithm method can be used with slight adaptations. It is a simple matter to rerun the genetic algorithm several times with the appropriate scoring system and to choose a parameter set from those produced.

The next chapter takes this small working cell assembly and incorporates it into networks of ever increasing complexity.

# Chapter 4

# Experiments on small associative memories

This chapter describes the initial work based on the model described in chapter 3 on page 53. Firstly, it is established that a cell assembly can be created in a small network and that it demonstrates the properties listed by Sakurai (section 1.1 on page 4) as being the essential properties of cell assemblies. Then larger networks are constructed in which a number of primitives can be set up and connected in such a way that they form compound assemblies. The properties of these compound assemblies are then investigated. Section 4.8 on page 106 describes experiments performed to determine the effect of spontaneous activation of cells on the size and reliability of cell assemblies. Spontaneous activation encourages dynamic growth and reduction in cell assemblies, and may prove to be a useful tool in reconfiguring networks of competing assemblies to improve overall performance.

The simplest way to construct cell assemblies is to specify the strengths of the weighted connections in advance to suitable predetermined values. These have been established already by a genetic algorithm, described in section 3.4 on page 65. However, one of the strengths of Hebb's original cell assembly theory is that the assemblies can come into existence as a result of connections adapting themselves in response to cells co-firing. The ability to learn new cell assemblies is therefore a powerful element of cell assembly theory. For this reason, an experiment is described that allows weight strengths to be learned, and this establishes that a simulated network of cells

can indeed learn simple cell assemblies. Later experiments did not include learning simply due to pressure of time and resources. There is no reason, in principle, why hierarchies of cell assemblies cannot be learned in a similar way to small networks of cell assemblies.

One of the criticisms that Milner [134] levelled at the cell assembly concept is that he could not understand how the connections within cell assemblies differed from those linking assemblies representing linked concepts. He implied that such assemblies would merge into one large assembly. Hebb [75] provided a possible solution to this, stating that cell assemblies linked within a concept can also be ignited independently, indeed they may take part in several other linked concepts. In this way, connections within assemblies differ in strength from those between assemblies. The experiments carried out in this chapter demonstrate that this is indeed the case. The learning process automatically establishes stronger connections within assemblies than between them, enabling them to be ignited independently of each other and any super-grouping of assemblies.

The network model implemented in these experiments was the one determined in chapter 3 on page 53. Initial experiments prove that isolated cell assemblies, the so-called primitive cell assemblies, can exist in the network of cells, and that they demonstrate the essential properties of cell assemblies as listed by Sakurai [154]. Further experiments show that primitive cell assemblies can participate in groups of primitive cell assemblies, termed 2-3 cell assemblies, as any two participating primitive cell assemblies are enough to activate the third. The last set of experiments shows how one 2-3 cell assembly interacts with others and how problems can arise due to erroneous activation of primitive cell assemblies.

One other important aspect in which the simulated cell assemblies behave is also investigated, specifically the effect of spontaneous activation of cells on size of cell assemblies. It is believed that cells in the cortex sometimes fire spontaneously rather than as a result of incoming signals from other cells [170, 188]. Experiments are described showing that spontaneous firing of cells on the edge of an existing cell assembly coupled with learning allows the cell assembly to expand in size. Such an effect can also be responsible for the dissolution of cell assemblies, roughly analogous

to forgetting, or cell assemblies splitting into two, which may be thought of as a concept becoming more specialised ("animal" into different types of animal such as "dog", "dog" into different breeds of dog). Such splitting has some bearing on the possibility of hierarchies of cell assemblies, the other area of research described in this thesis, as the resulting splinter cell assemblies originally had strong mutual connections, and yet can now be ignited independently.

One problem encountered is that of determining whether a particular cell assembly is activated or not. The activity of a cell assembly may be determined easily by counting the number of cells within that cell assembly that fire at any particular time step, but experiments show that cell assemblies tend to follow a continuum of activation, from complete inactivity in which no cells fire, to highly active in which, typically 40% of the cells fire at any particular instant. At what point should the cell assembly be declared active? There is no one correct answer to this question, but in order for the success or failure of an experiment to be determined, a threshold must be decided upon. It was found in experiments described in later chapters that primitives that did not ignite still demonstrated a small amount of activity, with up to 4 or 5 cells firing at any time step. For this reason, I have chosen to set an activation level of 10 of the cells within a primitive, *i.e.* if 10 or more of the cells comprising any cell assembly fire at any time step, that cell assembly is deemed to be active. This gives a small margin of error over the "noise" level present in inactive primitives.

## 4.1 One primitive Cell Assembly

The experiments described in this section demonstrate that the parameters derived by the genetic algorithm (section 3.4 on page 65) are indeed sufficient to sustain activity in a network of cells. To achieve this, the network had to sustain activity in at least 10 of its cells for at least fifty time steps. Early experiments were carried out using networks of 400 cells, but these were later repeated using 150 cells, with a view to scaling experiments up later. Although a primitive of 400 cells can easily be simulated in a personal computer, networks with many primitives the same size

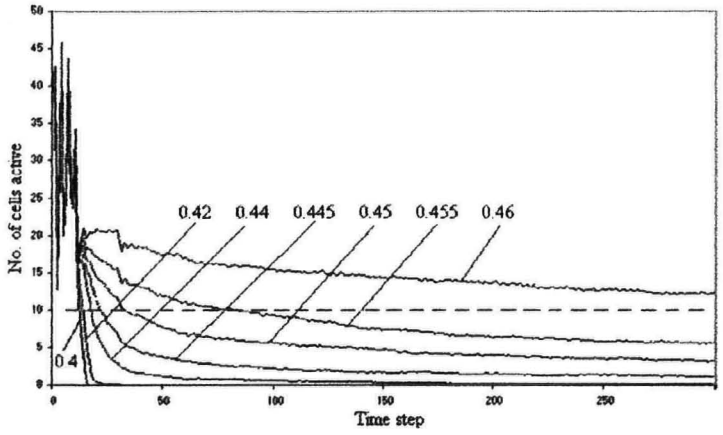


Figure 4.1: Network activity for one primitive cell assembly for different excitatory weight values (labelled). The dotted line shows the 10 cells level above which the cell assembly is considered active.

would require a great deal of time and hence be impractical. The results for 400 cells essentially duplicated those for 150 cells, reported here.

### 4.1.1 Establishment of a single assembly

An experiment was carried out on a network of 150 cells in which one primitive cell assembly was developed. The parameter values, determined by genetic algorithm, are shown in table 4.1 on page 77, with the weights from inhibitory cells having been determined automatically by those from the excitatory ones (section 3.2 on page 58). There were, of course, no connections between cells in different cell assemblies. Figure 4.1 shows the mean number of cells firing in the network for each of 300 time steps, for each of several different excitatory weight values. In each case the inhibitory weights were the negative equivalents of the excitatory weights.

Figure 4.1 shows us that cell assembly activity follows a continuum, with excitatory weights above 0.4 leading to cell assembly activity that persists for lengths of time that increase with excitatory weight. Extending the experiment beyond 300 time steps shows that for weights below 0.46, all cell assembly activity does eventually die out, whereas excitatory weights above 0.46 lead to activity that persists even beyond 300 time steps. It should be emphasised that these results are not general

- they apply only to this part of the parameter space. An entirely different part of the parameter space may well give entirely different results. Coincidentally, the most appropriate excitatory weight is reasonably close to the weight value predicted by the cell co-firing rate of 40% (section 3.2 on page 58).

It is interesting to compare the excitatory weight value for this experiment with that in table 3.1 on page 67. 0.44 is considerably lower than 0.5. The value of excitatory weight strengths differs depending on the exact circumstances of the experiment. In this case, there is no danger that one cell assembly will accidentally ignite another. The weights given in table 3.1 assume a 2-3 compound assembly. The high excitatory weight strength for connections within a primitive occur due to the "lucky neuron" effect (see section 4.6 on page 95) by which a few cells within a primitive fire and are responsible for igniting the assembly as a whole. 0.44 is the optimal figure to give assembly activity that dies out after a few tens of time steps. This restriction had to be dropped for later experiments, particularly those involving large number of primitives, in which the problem of finding suitable parameter values become much harder.

While 0.44 is low compared to 0.5, it is quite high when compared to the preferred value of 0.08 in experiment 4.2 on page 80. In experiment 4.2, a network of three assemblies were connected in such a way that activating any two primitives is sufficient to ignite the third but activating only one is insufficient. Such a configuration puts extra constraints on the range of the weights between primitives.

The weight strength is also influenced by the number of connections between the cells. The genetic algorithm in this case preferred 6 connections between cells, relatively low compared to numbers of connections in later experiments. Loosely speaking, one can consider a total signal level passing between cells within a cell assembly, which is approximately proportional to the product of the number of connections and the mean weight between cells, *i.e.* proportional to $c(EW + (1 - E)w)$. This signal level can be maintained as the number of connections between cells is reduced by increasing excitatory weight strength and/or moving the inhibitory weight strength towards 0. However, experiments show that, in a single cell assembly that ignites reliably, the relationship between $c$ and $W$ is not one of simple inverse propor-

| Variable | Variable name | Default value |
|---|---|---|
| $c$ | Number of connections from each cell | 6 |
| $P$ | Probability of any cell being activated externally on any given time step | 0.4 |
| $T$ | Total number of time steps involving external activation | 10 |
| $N$ | Total number of cells (per primitive) | 150 |
| $N_1$ | Number of cells that fire on average per time step during first $T$ time steps | 60 |
| $E$ | Proportion of cells that are excitatory | 0.8 |
| $W$ | Weight to each destination cell from each excitatory cell | 0.44 |
| $w$ | Weight to each destination cell from each inhibitory cell | -0.56 |

Table 4.1: Definition of variables with default values.

tionality, and illustrate that the relationship between the parameters is a complicated one.

Figure 4.1 does show two interesting features. Firstly, in the first ten time steps, the number of cells that fire fluctuates wildly. Investigation shows that this is a side-effect of fatigue. The external activation of the cell assembly is intense and causes a large number of cells to fire (although not 40% of the cells as might be expected). The fatigue of these cells increases by 0.19, which prevents the cells firing the next time, as the activity minus the fatigue cannot exceed the 0.95 firing threshold. Cells recover at roughly half the rate at which they fatigue, so after a further two time steps, a cell has recovered sufficiently to fire. This suggests that the pattern of firing in the first two time steps should follow a roughly repeated pattern with frequency three time steps. Figure 4.2 shows the first ten time steps of figure 4.1 in detail, in which such a pattern is clear. Experiments show that varying the ratio between fatigue rate and recovery rate does result in patterns of different frequencies appearing, although different ratios do not generally permit persistence of activity beyond the period of

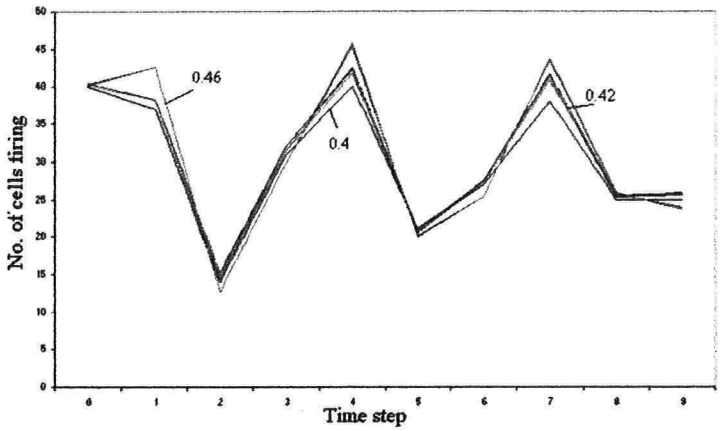


Figure 4.2: First ten time steps in detail. The pattern of activation is almost identical for a variety of excitatory weight strengths between 0.4 and 0.5.

initial stimulation.

Additional experiments showed that adjusting fatigue rate and recovery rate could lead to other frequency patterns appearing. For instance, a fatigue rate and recovery rate of 0.21 and 0.07 respectively led to a peak in the cell activity approximately every fourth time step. Of course, there are limits to the frequency patterns that can be set up, not only because there are only 10 time steps in which they can manifest themselves, but also because too high a fatigue rate or too low a recovery rate prevents reliable cell assembly activation. If too many cells fatigue on any time step, there are not enough available to fire on the next time step and cell assembly activation risks dying out all together. Extending $T$ showed that the frequency pattern generally persisted, but $T$ was generally kept at 10 time steps for reasons explained in section 4.2 on page 80.

There is little point in reporting a percentage success rate for this experiment, since it is a trivial matter to adjust parameters so that the single assembly ignites on every trial. It is not sufficient, however, simply to raise the excitatory weights between cells to their maximum value. Such a strategy simply cause large numbers of cells to fire, and then fatigue simultaneously. The small number of cells capable of firing on the next time step is usually incapable of sustaining sufficient activity to persist while the fatigued cells recover. However, on a few occasions, the "lucky

neuron" effect (section 4.6 on page 95) does permit activity to persist.

### 4.1.2 Can weights be learned?

The next experiment investigated whether weight values could be learned. In this experiment, all excitatory weights were initialised to random values between 0 and 0.4, and all inhibitory weights to random values between -0.4 and 0. The external stimulus was identical to that in Experiment 4.1.1 on page 75. Hebbian learning took place at the end of each time step for all time steps in each program run. Parameter values were identical to those in Experiment 4.1.1, with the exception of the weight values, of course.

Figure 4.3 shows the average weight values, both excitatory and inhibitory, after each time step, with weights being initialised to zero in all cases. As explained in section 3.2 on page 58, activating cells with a probability of 0.4 encourages excitatory weights to tend towards 0.4 and inhibitory ones to –0.6, the negative equivalent weight, although this trend ignores the possibility of fatigue, which reduces the probability of cells firing together, and reverberation, in which cells activate others thereby increasing the excitatory weight between them. The results show that weights do indeed converge approximately on the predicted values. In fact, increasing the number of time steps shows that the weights converge on approximately 0.48 for excitatory weights and -0.55 for inhibitory weights, and the difference between these and the predicted values may be ascribed to the reverberation effect. Another possible cause is the lucky neuron effect.

Experiments 4.1.1 on page 75 and 4.1.2 show that a simple primitive cell assembly can exist within the network of cells specified, and that it fulfils the general prediction about the weight values. Parameter values in a single cell assembly are less rigidly controlled than in systems containing more than one assembly, as the only purpose of the assembly is to ignite. Provided the excitatory weights are sufficiently high, this is almost guaranteed to happen. Too high a weight does cause activity to persist indefinitely, but there is no theoretical reason why assembly activity should not do this. The next stage is to start combining primitive cell assemblies to investigate whether they can form compound 2-3 cell assemblies, the simplest form of compound

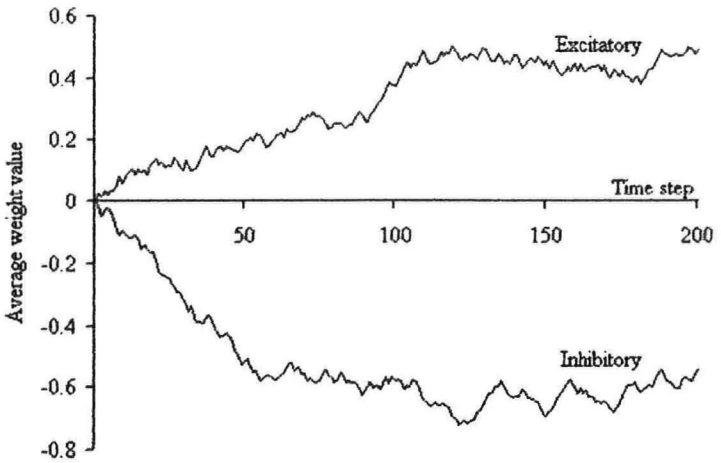


Figure 4.3: Adaptation of weight strength with time. The graph shows the mean value of weights from excitatory and inhibitory cells over 100 different training runs after given numbers of time steps. All weights were initialised to 0.

cell assemblies.

## 4.2 Three primitive cell assemblies

The experiment described in this section investigated whether a combination of two primitive cell assemblies could be used to activate a third, with each being unable to activate the third individually. Such a combination represents the simplest possible compound cell assembly, as any simpler one could not be considered to be a combination of primitives. A combination of two primitives in which activating one was sufficient to ignite the other would in effect be one single primitive cell assembly. Hebb [75] (p. 105) specifically stated that cell assemblies should behave in this way, with one single assembly being incapable of igniting another except in the presence of a third.

This problem is substantially harder than the one in experiments 4.1.1 and 4.1.2 (on pages 75 and 79 respectively). If excitatory weights between cells in different primitive cell assemblies are too low, a combination of primitive cell assemblies will not provide enough activation energy to activate the third. However, if weights are too high, activating a single primitive cell assembly will lead to runaway activity

| | |
|---|---|
| Weights from excitatory cells within a primitive cell assembly | 0.5 |
| Weights from inhibitory cells within a primitive cell assembly | -0.5 |
| Weights from excitatory cells between primitive cell assemblies | 0.08 |
| Weights from inhibitory cells between primitive cell assemblies | -0.92 |

Table 4.2: Weight settings for the $ABC$ 2/3 network.

throughout the network. Such uncontrolled activity may be thought of as analogous to epilepsy in the human brain.

A network of 450 cells was used, with 150 assigned randomly to each primitive cell assembly. Global parameter settings were the same as those for experiment 4.1.1 with the exception of the number of connections per cell and the weight strengths. Since the number of primitives had increased from one to three, the number of connections had to increase as well to maintain the general level of connectivity. A simplistic approach would involve maintaining all parameters at the same level as in experiment 4.1.1 and just scaling up the number of connections. Another approach is to use a genetic algorithm to determine the optimal values of the weights (see section 2.4 on page 48), with the scoring function described in section 3.4 on page 65. The genetic algorithm revealed that setting the weights from excitatory cells to others within the same primitive cell assembly to 0.5 (and the corresponding weights from inhibitory cells to -0.5), it was found that the number of connections could be kept at 20 per cell. This produced more reliable results than increasing the number of connections per cell.

The best weight settings that were obtained are shown in Table 4.2.

Figure 4.4 shows the results of one trial run of Experiment 4.2, which involved the activation of just one primitive cell assembly. It shows that the activated cell assembly maintains a roughly constant level of activity, while not producing any substantial activity in the other primitive cell assemblies. Extending the number of time steps beyond 300 shows that the number of cells firing gradually decreases, reaching zero after approximately 2000 time steps. Figure 4.4 shows that the activity levels of the activated primitive fluctuate rapidly as the cells fatigue and recover, although analysis using MATLAB was unable to find an underlying frequency pattern to these

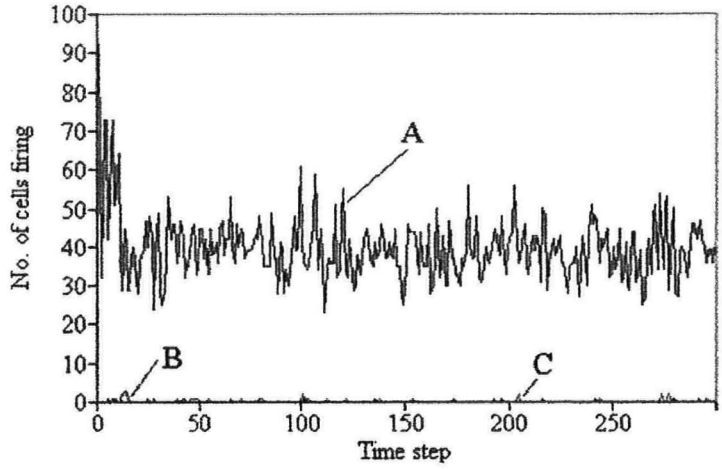


Figure 4.4: Result of activating a single primitive cell assembly ($A$ in this case) out of 3. Although there is trace activity in the other primitive cell assemblies ($B$,$C$), they remain essentially inactive. Activating $B$ only or $C$ only produces a similar graph.

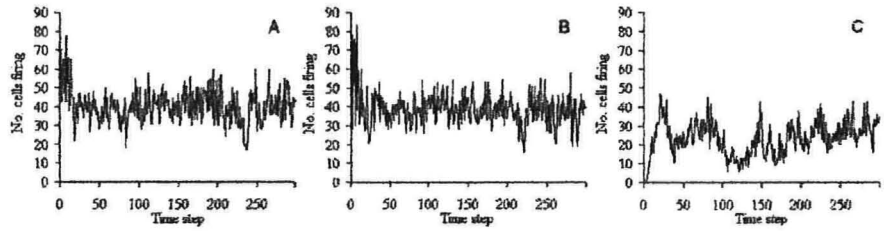


Figure 4.5: Result of activating 2 primitive cell assemblies ($A$ and $B$ in this case) out of 3. $C$ activates within a few time steps and maintains roughly the same activation level as A and $B$. Activating $A$ and $C$, or $B$ and $C$ produces a similar graph.

fluctuations. Experiment 4.2 was repeated 10,000 times and it was determined that the average number of cells that fire per time step was approximately 40 for the first 100 time steps. Figure 4.5 shows a run in which two primitives provided enough activity to ignite the third. Repetition of the experiment shows that this is a general result. With the weights set as in table 4.2, activating any two primitives is sufficient to ignite the third, but activating one primitive is not. The different possible outcomes for each run are shown in table 4.3.

Figure 4.6 shows the compromise chosen as regards the two most important parameters governing the behaviour of the network, *i.e.* the number of connections

One primitive active:

| | |
|---|---|
| Primitive remains active | 96.7% |
| Primitive dies out quickly | 1.4% |
| Other primitive(s) ignite | 1.9% |

Two primitives active:

| | |
|---|---|
| All three primitives active | 99.3% |
| Third primitive fails to ignite | 0.5% |
| All activity dies out | 0.2% |

Table 4.3: Outcomes of experiment 4.2

per cell and the excitatory weight strength between cells. Large numbers of connections and/or strong weights encourage two activated primitives to ignite the third, whereas low numbers of connections and/or weak connection strengths discourage a single activated primitive from igniting either of the others. Figures 4.6(a) shows the probabilities of one single active primitive remaining active while not activating another for given values of each of the two parameters near the optimum chosen by the genetic algorithm. Figure 4.6(b) shows the probability of two primitives igniting the third for the same part of the parameter space.

Experiments showed that the most critical two parameter setting were the weights from excitatory cells to those in other cell assemblies and the number of connections per cell. Figure 4.6 shows the result of varying these parameters around the default values shown in Table 4.2. These figures illustrate that the relationship between these two factors is a complicated one. The probability that activating one primitive cell assembly will cause activity to persist only in that cell assembly has a maximum when the number of connections is about 19 provided that the excitatory weights are not greater than about 0.06. With a greater excitatory weight or a greater number of connections, activating the primitive cell assembly causes saturation in the entire network. With fewer connections per cell, the primitive cell assembly is incapable of sustaining activity. However, the picture is simpler when considering activating two primitive cell assemblies. In this case, saturation is the desired condition, and the probability of it happening increases with increasing number of connections and

increasing excitatory weight. The parameter values used in the other experiments represented a compromise condition - slightly sub-optimal in both cases. A rough estimate of the optimal parameter settings can be found simply by multiplying the two probabilities, as shown in figure 4.6(c). The graphs show a fairly rapid drop in the success rate in each of the two firing situations if one moves away from the optimal parameter settings in almost all directions, indicating that the system is not robust as regards these parameter settings. A slight ridge in the parameter space is evident, favouring low numbers of connections with high excitatory weights and vice-versa. The optimal setting is close to the centre of figure 4.6(c), although it does not offer much of an improvement above the regions immediately contiguous to it.

It was found that the reliability of assembly ignition depended on the value of $T$, the number of time steps for which external activation was provided. For $T \gg 10$ activating two primitives externally caused the activity in the third to peak and then die out rapidly in a significant number of cases. This effect was attributed to certain crucial cells in the destination primitive (so-called "lucky neurons", section 4.6 on page 95) being repeatedly stimulated to the point where they could not recover rapidly enough from fatigue to maintain assembly activity. This effect was stochastic, but the number of lucky neurons was sufficiently large in any destination assembly to make it a considerable problem. This was one reason for choosing $T = 10$ time steps.

## 4.3 Five primitive cell assemblies

The next stage of complexity is to construct a network containing five primitive cell assemblies, termed $A$, $B$, $C$, $D$ and $E$. There is little point in constructing a network with four primitives since such a network could only support one 2-3 compound assembly. Parameters remain the same as for previous experiments - there is no reason to change them - except that the number of connections per cell is raised from 20 to 34, an increase in the approximate ratio 5 : 3. This allows the same average number of connections from one primitive to another as the number of primitives has been increased. It was found that such an increase allowed one single primitive cell

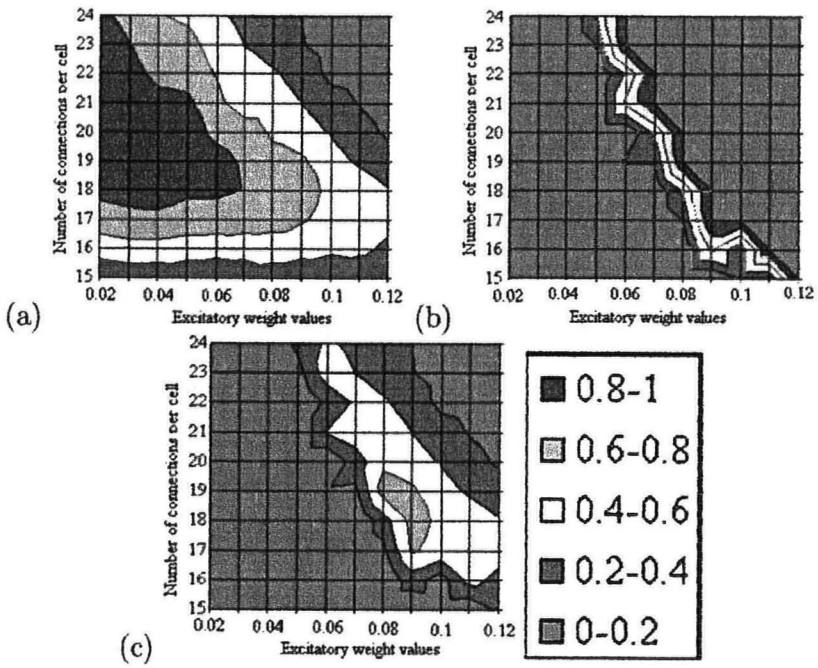


Figure 4.6: Varying weight strength from excitatory cells to those in other primitive cell assemblies. (a) shows the probability that activating any single primitive cell assembly results in activity in that cell assembly persisting and no other primitive cell assemblies are activated. (b) shows the probability that activating two primitive cell assemblies results in the third becoming activated. Both of these outcomes constitute success in terms of the experiment. (c) shows the product map of the two grids giving an indication of suitable combinations that are most likely to lead to success in both cases. The probability ranges are shown by the key.

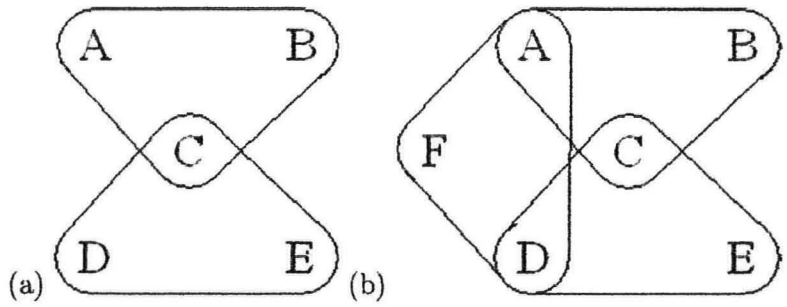


Figure 4.7: Possible configurations for (a) five primitives and (b) six primitives.

assembly to activate two others erroneously, unless the excitatory weight between cells was reduced to 0.06, with the inhibitory weight strength being the negative equivalent.

It is a trivial matter to set weights between cells in order to implement two 2-3 cell assemblies, e.g. $ABC$ and $CDE$, which must share one primitive cell assembly in common, as shown in figure 4.7(a). Each 2-3 cell assembly obeys the same principles as derived in section 4.2 on page 80, *i.e.* activating only one cell assembly fails to activate the entire 2-3 cell assembly, but activating any two of the three is sufficient. Other researchers have shown that multiple cell assemblies can exist in which cells participate in more than one assembly [91].

However, additional issues arise concerning the interaction of the 2-3 cell assemblies. Cell assemblies $A$ and $B$ should inhibit the activation of $D$ and $E$ and vice-versa in order to avoid saturation throughout the entire network, so the inhibitory connections between them should be set to the minimum value (-1) and the excitatory ones to 0. The question then arises as to which cell assemblies would activate if some unexpected combination, such as $A$ and $D$, were stimulated. One possibility is that one of these would de-activate the other, another is that they would activate $C$, the common cell assembly, and that would lead to the activation of either $ABC$ or $CDE$.

Experiment 4.3 was carried out in order to confirm the feasibility of two overlapping compound cell assemblies. A network of cells was constructed, 150 cells to each primitive cell assembly. Each cell was given 35 connections in order to ensure that the average number of connections between the cells in each primitive and those in any other primitive was approximately the same as in previous experiments. Excita-

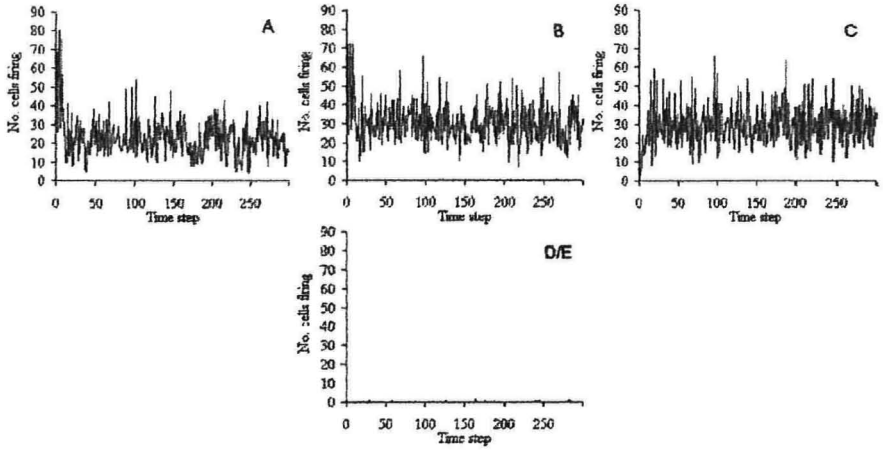


Figure 4.8: Activating primitives $A$ and $B$ in a network containing primitives $A$ to $E$ causes $C$ to activate reliably and negligible activation in $D$ and $E$. Similar results occur when $D$ and $E$ are activated.

tory weights were 0.5 between cells in the same primitive cell assembly, 0.06 between cells within the same compound cell assembly (e.g. between cells in primitive $A$ and those in primitive $B$), and 0 between cells in unrelated primitive cell assemblies (e.g. between cells in primitive $B$ and those in primitive $E$). Inhibitory weights were set accordingly: –0.5 between cells in the same primitive, –0.94 between cells in the in same compound cell assembly, and –1 between cells in unrelated primitive cell assemblies. The results of experiment 4.3 are shown in figure 4.8.

However, an interesting result occurs when two conflicting primitives are activated. Figure 4.9 shows a typical result when primitives $A$ and $D$ are activated. Since these primitive cell assemblies are not present within a 2-3 cell assembly, they have strong inhibitory mutual connections. It is expected that one of these primitives should tend to shut the other one down. In this particular case, $D$ reduces the activity in $A$, although repeated experiments indicate that each outcome is equally likely. However, figure 4.9 shows that the primitive cell assembly connected to both activated ones, namely $C$, is rapidly ignited. This often leads to the completion of one 2-3 cell assembly, which in turn shuts down the rogue primitive cell assembly more quickly. In this experiment, $A$ and $D$ both provide half the activation that $C$ requires to ignite. In principle, this could either ignite $ABC$, or $CDE$. However, by

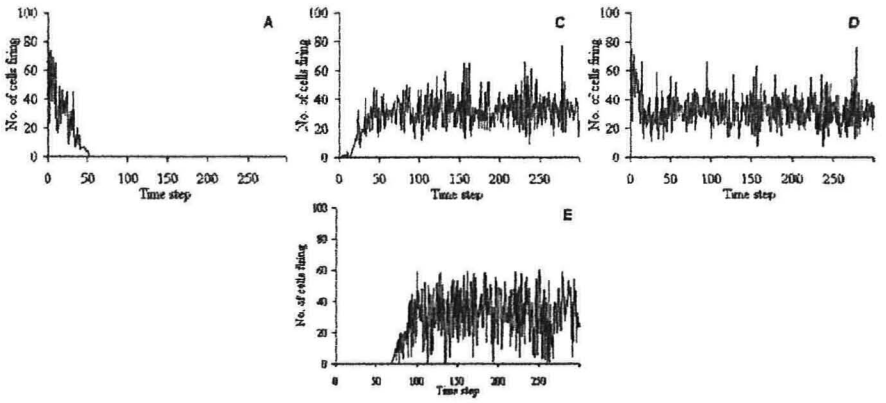


Figure 4.9: Activating two conflicting primitive cell assemblies, here $A$ and $D$, causes one to be suppressed, but not before activating the primitive cell assembly to which they are both connected, $C$. This leads to the completion of 2-3 cell assembly $CDE$. $B$ remained completely inactive throughout this experiment, and has not been plotted in order to save space.

this stage, $D$ is more active than $A$, so $CDE$ often ignites. The presence of $D$ and $E$ together shut $A$ down completely within a few time steps of $C$ igniting.

Figure 4.10 shows the result of activating the mutually connected primitive cell assembly and any one other, e.g. $C$ and $E$. In this case, $D$ is activated almost immediately, and $A$ and $B$ not at all. Although $A$ and $B$ receive half the necessary activation, the presence of both $D$ and $E$ are enough to suppress them completely. In many ways, this experiment echoes the one in which $A$ and $B$ were activated. The results are the same, with one 2-3 cell assembly active and no activity in the other primitive cell assemblies.

## 4.4 Six primitive cell assemblies

Experiment 4.3 on page 84 showed that a network of five primitive cell assemblies could be constructed, and could contain two compound cell assemblies. In this section, I demonstrate that increasing the number of primitives adds problems in the form of erroneous ignition of compound cell assemblies.

In experiment 4.4, an extra primitive, termed $F$, was added to the network and the

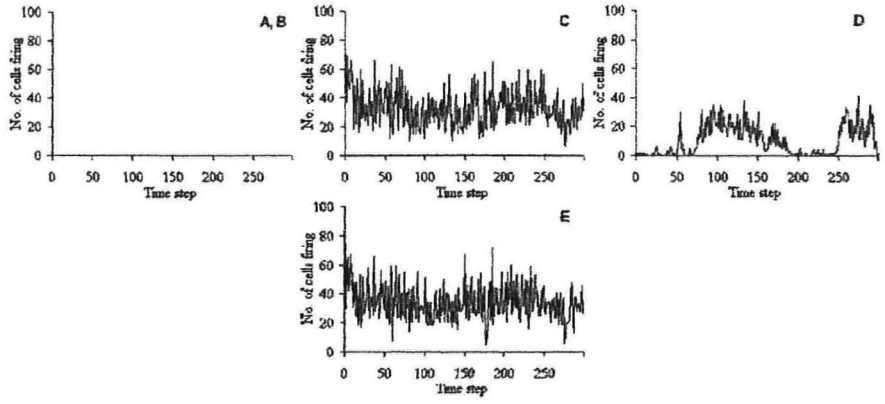


Figure 4.10: Activating $C$ and $E$ causes the activation of $D$ but no activation of $A$ and $B$. A similar effect occurs when $C$ is activated with any other primitive cell assembly.

number of connections per cell was scaled up to 40. 2-3 cell assemblies $ABC$, $CDE$ and $ADF$ were constructed, as shown in figure 4.7(b) (page 86). This combination of compound assemblies was chosen as no combination of two primitives appears in more than one compound assembly. Parameters were kept as in the previous experiment. The experiment was carried out to test the different possible outcomes of the network containing six primitives. The situations tested were as follows, with persistence being defined as activity above the 10 cell threshold for a chosen time of at least 30 time steps beyond the point where external stimulation is removed. Each numbered situation involved 10,000 trials, with suitable externally activated primitives being chosen at random (e.g. for situation 1, in which a single primitive is activated externally, each primitive had an equal probability of being chosen.)

1. One primitive is activated externally. Success occurs if the activity in the primitive persists without igniting any other.

2. Two primitives present in a 2-3 assembly are activated externally. Success occurs if they ignite the third without any erroneous ignition.

3. Two primitives not present in the same 2-3 assembly are activated externally. Success occurs if either one shuts the other down, or if activity in both dies out.

| Situation tested | Success rate |
|---|---|
| 1 | 97.14% |
| 2 | 51.10% |
| 3 | 99.13% |
| 4 | 23.07% |

Table 4.4: Success rates for networks of six primitives for weights determined by genetic algorithm.

4. Two 2-3 compound assemblies sharing a single primitive are activated externally (e.g. $ABC$ and $CDE$ sharing primitive $C$). Success occurs if one of the cell assemblies persists but the other does not, and if the sixth primitive ($F$ in this case) does not ignite.

It was confirmed that any single primitive cell assembly maintained its own activity when activated, but was in the vast majority of cases incapable of activating any others, and that, in general, two primitive cell assemblies of a 2-3 cell assembly did activate a third to form a stable 2-3 assembly, although the assembly ignited was not always the expected one. The exact success rates are shown in table 4.4.

The large drop in the success of the system for situation 2 can be explained as follows. Primitive cell assemblies in a 2-3 cell assembly activate when they receive half the necessary energy from each of the other two participants. Although it might seem that this energy is delivered by a range of connections from the contributing primitives, in practice, it involves the firing of only a few neurons, which I term "lucky neurons" (section 4.6 on page 95). When $A$ and $D$ are activated, $F$ receives enough energy to activate, as intended in $ADF$. However, $C$ receives half the activation energy from $A$ (in $ABC$) and the other half from $D$ (in $CDE$). There are strong inhibitory links between $C$ and $F$, since they are not within a pre-programmed 2-3 cell assembly, so we would expect either one of them to shut the other one down. In practice, the one that ignites first simply prevents the other igniting in the first place. This means that activating $A$ and $D$ leads to a 2-3 cell assembly, either $ADF$ (with no activity in $C$) or $ACD$ (with no activity in $F$), with each outcome being approximately equally likely (see Table 4.5). Figure 4.12 shows two typical program

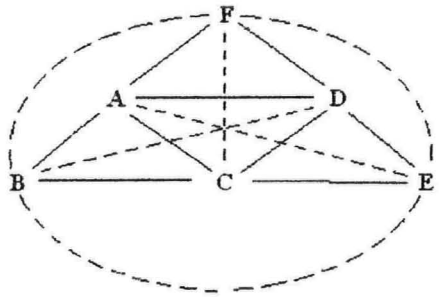


Figure 4.11: The excitatory and inhibitory connections for an $ABC$, $CDE$, $ADF$ arrangement. Excitatory connections are shown by solid lines, inhibitory ones by dashed ones.

runs, in which both these occur. However, the graphs shown in figures 4.12(a)-(d) are an example of a "clean" run, in sense that no further ignition occurs. In approximately 15% of cases in which $C$ is erroneously ignited, further ignition does occur.

The combination $ACD$ does not represent a designated cell assembly, and this can lead to further change. The tendency for $A$ and $C$ to ignite $B$ can lead to a temporary state in which $A$, $B$, $C$ and $D$ are all active. The strong mutually inhibitory connections between $B$ and $D$ invariably lead to activity in one of them dying out. The final result of activating $A$ and $D$ is therefore one of three possible 2-3 compound assemblies: $ABC$, $ADF$ or $ACD$, the last of which is not even designated by the connections. Although a complicated sequence of primitive activation occurs when $C$ ignites in this case, the activation pattern when $F$ ignites is more straightforward. In this case, there is no erroneous ignition.

The relationships between the primitives is best summarised in figure 4.11, in which solid lines represent excitatory connections and dashed lines inhibitory ones. The 2-3 assemblies are represented by the triangles of solid lines. The figure shows that the unintentional assembly $ACD$ has been stored. Activating $A$ and $D$ can therefore ignite $C$ or $F$, but is unlikely to ignite $B$ or $E$ directly. The inhibitory connections mean that $ABC$ or $CDE$ are unlikely to be ignited, although this may happen as a result of further interactions between primitives.

A similar problem occurred when $A$ and $C$ were activated. In this case, either

| Outcome | Percentage occurrence |
|---|---|
| $ABC$ | 12.22% |
| $ADF$ | 48.71% |
| $ACD$ | 38.91% |
| Neither $C$ nor $F$ activated reliably | 0.16% |

Table 4.5: Percentage likelihood of possible outcomes when $ABC$, $CDE$ and $ADF$ are present in a network and $A$ and $D$ are activated. This can lead to one of three compound assemblies being activated, one of which is not even designated by the connections. The figures are based on 10,000 trials.

$B$ or $D$ activated, due to the connections in $CDE$ and $ADF$. Although activating two related primitives did invariably lead to a stable 2-3 compound assembly being ignited, the uncertain identity of the third primitive led to approximately half of the situation 2 trials being classified as failures. This continual ignition and suppressing of primitives does, however, open up the possibility of sequential information processing, in which primitives "battle it out" to reach an optimal solution to a problem, in a similar manner to the Sharks and Jets network described in chapter 1 on page 2.

A related problem occurred when $A$ and $F$ were activated. In this case, $D$ activated, because of the connections in the cell assembly $ADF$, which led to primitive $C$ receiving some activation. However, in the great majority of cases, activity in $C$ did not build up to appreciable levels, as $F$ was active and fully established by that point.

Situation 4 has a surprisingly low success rate. It was found that activating five out of the six primitives almost always led to runaway activity in the entire network followed by primitives dying out in an unpredictable way. It is reasonable to infer that the excitatory contributions from five of the primitives to the sixth outweigh the inhibitory contributions from fewer than five primitives. This usually leads to the ignition of the sixth primitive within a few time steps.

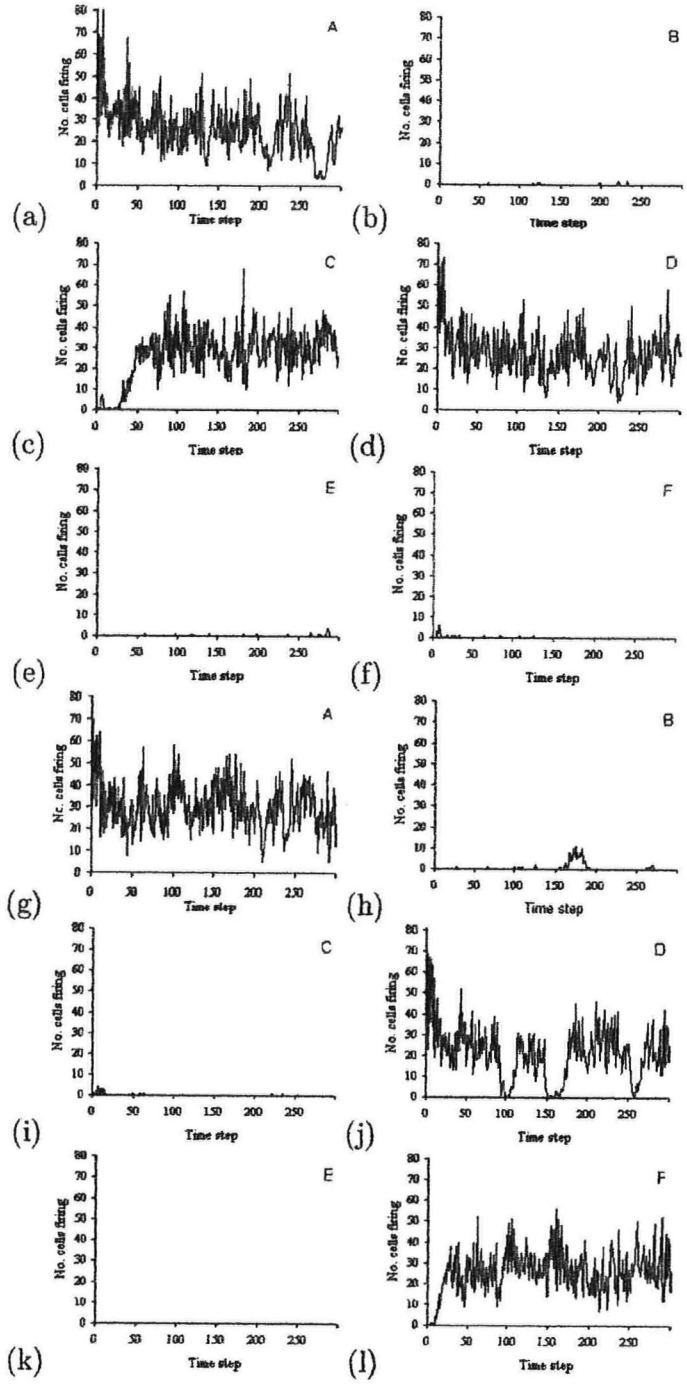


Figure 4.12: Activating $A$ and $D$ leads either to $C$ activating (a-f), or $F$ (g-l). In either case, activity in the other primitive cell assembly is suppressed. There is a small peak in both $C$ and $F$ during the first ten time steps, followed by a rapid decline in one of them.

| Outcome | Percentage occurrence |
|---|---|
| One active primitive persists alone | 20.09% |
| Erroneous ignition in other primitives | 79.91% |

Table 4.6: Increasing internal excitatory weight strengths from 0.5 to 0.52 causes single primitives to persist, but also allows erroneous ignition of other primitives.

## 4.5 Assembly persistence

One of the essential criteria of a simulated cell assembly is that activity should persist beyond the point where the external simulation is removed. This is easy to achieve in networks containing one single assembly as internal excitatory weight strengths can be set high enough to allow activity to persist indefinitely. Indeed, connections can be set to give a wide range of persistence times, although it is not a simple matter to predict the persistence time from the parameter settings.

The problem of persistence only becomes an issue when more than one primitive assembly interacts. It is still possible to set connections for an assembly to persist for a long time, but extended persistence increases the likelihood of erroneous ignition of other primitives. It is perfectly possible that the main reason that primitives are not ignited by the wrong combination of active primitives is that their activity dies before erroneous ignition is possible. The parameter set used gives assemblies and combination of assemblies that persist for between 50 and 250 time steps after the external stimulus has been removed, with the mean value being about 150 time steps. Given the same scale as in Figure 2.6, 150 time steps corresponds to about 1500ms. Figure 4.1 on page 75 shows that increasing the excitatory weight values within a primitive can be used to increase the persistence of a cell assembly, so in order to test the long-term effects of assembly activity, the weight was increased from 0.5 to 0.52. It was found that in a substantial number of trials, one active primitive was sufficient to cause ignition of others after approximately 200 time steps. Results are shown in table 4.6. When two primitives received external activation, erroneous ignition occurred in every single case.

The parameter set favoured by the genetic algorithm results in a network balanced on a knife-edge, between the death of activity in a primitive and uncontrolled ignition.

An analysis of the activity levels of cells indicates that there is a gradual build-up of activity in cells even in primitives that do not ignite. With excitatory weights of 0.5, the decay of each cell is sufficient to remove the build-up of activity before it tips the cell over into firing. With weights of 0.52, the cell is more likely to fire than not.

## 4.6 The lucky neuron effect

In the experiments described in previous sections, the excitatory weights between related primitives in a compound cell assembly are small. Generally the best results were obtained with weights less than 0.1. The inhibitory weights were set to the negative equivalent of this, i.e between -1 and -0.9. The magnitude of each inhibitory weight is therefore at least nine times as great as that of the excitatory ones. This means that although excitatory connections outnumber inhibitory ones by four to one, the average connection strength between primitives is negative. Experiments show that the total energy delivered by each externally stimulated primitive to each of the other primitives during the ten time steps of external activation has a mean value of -352 ($\sigma = 408$).

So why do the primitives activate at all? An experiment showed that the destinations of the connections from the inhibitory cells did not cover the cells in any primitive completely and evenly, and that a mean of 17.11% ($\sigma = 2.26\%$) of the cells activated received enough positive activation to fire when any one primitive was activated by two others, but that only 11.50% ($\sigma = 0.44\%$) received sufficient energy with only one contributing primitive. The high excitatory connections (0.5) of cells within a primitive were sufficient to sustain activity in the former case, but not in the latter case. 11.50% corresponds to approximately 17 cells, enough to declare a primitive ignited, but the fatigue rate to recovery rate ratio of 2:1 has the effect of cutting the average firing rate of these cells by a factor of 2. Curiously, the percentage of cells activating does not double when the number of contributing primitives rises from one to two, and this can be attributed to the fact that the larger number of connections more closely reflects the overall statistical average energy transfer, which is negative.

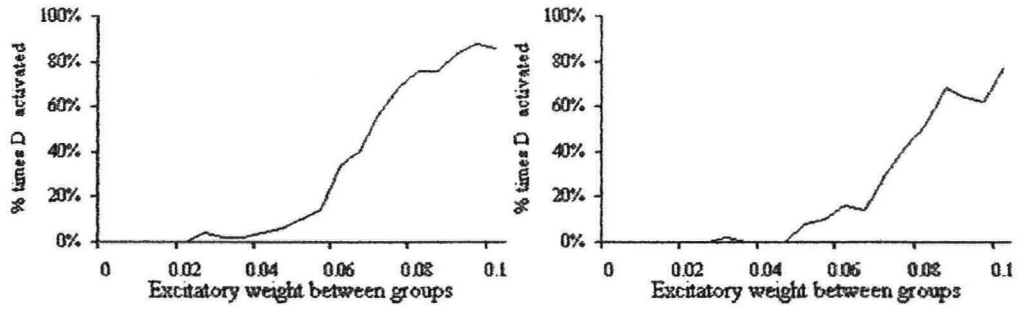


Figure 4.13: The probability that primitives $A$ and $B$ will activate $D$ increases as the strength of excitatory connections between cells in different primitives increases, as shown in (a). However, this probability decreases if a third primitive, $C$, also contributes, as shown in (b).

It therefore follows that three contributing primitives will exacerbate this effect, and this does turn out to be the case. When attempts are made to initialise a 3-4 cell assembly (*i.e.* primitives $A$, $B$ and $C$ are sufficient to activate $D$, but any two of $A$, $B$ and $C$ are insufficient) and small excitatory weight values (approximately 0.1 or less) are used for connections between primitives, then such an attempt fails. Figure 4.13 shows that, as the weight values increase, two contributing primitives (e.g. $A$ and $B$) have a greater probability of activating any other than three contributing primitives (e.g. $A$, $B$ and $C$).

Although this effect does allow 2-3 cell assemblies to be created, it does not bode well for 3-4 or higher order cell assemblies. Another set of parameters must be found in a different part of the parameter space. If we assume that the excitatory weights between primitives within compound cell assemblies is $x$, and that inhibitory weights reflect excitatory weights, then the inhibitory weights between primitives within the compound cell assembly is $-(1 - x)$.

The total activity delivered to any primitive by any two contributing primitives is given by equation 4.1 where the first term represents the excitatory energy, the second term the inhibitory energy, and $K$ is an appropriate constant. $K$ depends on number of cells and average activation. The coefficients take into account the fact that excitatory cells are 4 times more common than inhibitory ones. This rearranges to give $K(x - 0.2)$. Clearly, this becomes positive when $x$ is greater than 0.2.

$$\text{total activity} = 0.8Kx - 0.2K(1-x) \tag{4.1}$$

This does raise some questions about the parameter values derived from the genetic algorithm in section 3.4 on page 65. The genetic algorithm limits the excitatory weight strength between primitives to a narrow range of values centred around 0.08, which, according to equation 4.1 should lead to a total activation being passed between primitives that was negative. A statistical analysis of the activity between primitives showed that this was indeed the case. At first sight, it may appear that no primitive or combination of primitives can ever ignite any other given such a low excitatory weight strength, but this ignores the stochastic differences in the activation of cells in each primitive. The primitives are activated as a result of relatively few cells (typically between 10 and 20) firing. Although all excitatory connections between cells in different primitives had the same strength in the genetic algorithm experiment, destinations were distributed randomly, and it is perfectly possible for some cells within a primitive to be better connected than others. I term these "lucky neurons". An experiment in which energy transfer between primitives was limited to the 15 most highly connected cells in each primitive showed that a small amount of activation energy rapidly led to runaway ignition of all the cells in the network, indicating the powerful effect that these cells have. Figure 4.14 shows a typical run in which externally activating a single primitive ($A$) leads to a burst of uncontrolled activity in all primitives. This quickly leads to widespread fatigue, and the activity dies. This experiment suggests that the overall inhibitory effect of the connections between primitives is necessary in order to put a brake on the stimulation of primitives by the lucky cells within them. A further experiment was carried out in which all connections between primitives were left intact *except* those to the 10% most heavily connected cells which were lesioned. It was found that no primitive or combination of primitives could ever ignite any other one, thereby establishing without doubt that the lucky neuron effect is responsible for assembly ignition.

In experiment 4.4 on page 88 I referred to primitives being ignited by receiving half the necessary activation energy from each contributing primitive. This conclusion is not nullified by the discovery of the lucky neuron effect. Although the number of

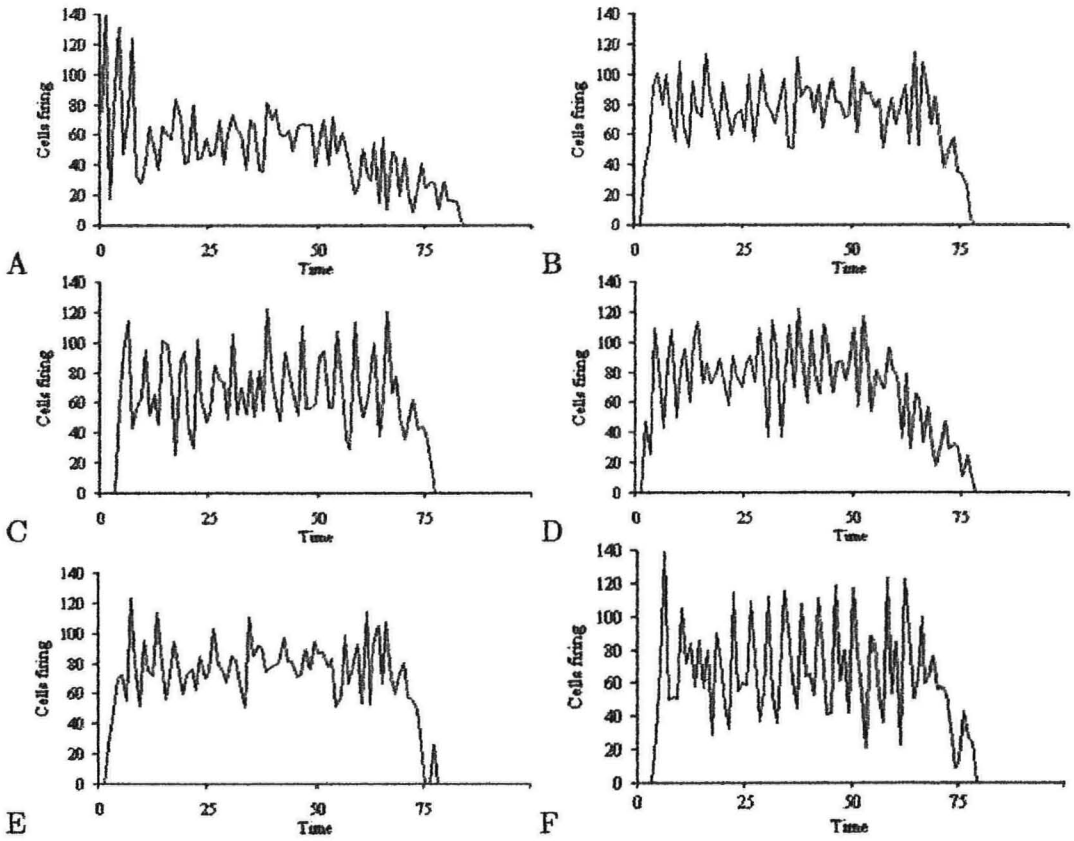


Figure 4.14: Restricting connections between primitives only to the 10% of cells with the most connections leads to uncontrolled activity followed by a crash as fatigue sets in. The number of cells firing in each primitive ($A$ –$F$) is shown.

lucky neurons firing does not double when two primitives contribute rather than one, the fact remains that one contributing primitive does not provide sufficient energy to ignite the primitive, whereas two contributing primitives do. The concept of contributing energy simply needs to be focussed onto the lucky neurons. It is insufficient, however, simply to say that doubling the energy contribution that the lucky neurons in a primitive receive simply doubles the number of lucky neurons that ignite. That clearly is not the case. Furthermore, fatigue also plays a part in reducing the number of lucky neurons that fire. However, it seems to be possible to arrange a network in such a way that primitives have a certain "global energy threshold", and that $k$ contributing primitives will provide enough energy to exceed this threshold whereas $k-1$ primitives will not. Experiments in this and the next chapter show that this is certainly the case for low values of $k$, although the problem becomes harder as $k$ increases.

An experiment was carried out to detect whether removing the lucky neuron effect changed the success rate appreciably. Experiment 4.4 was repeated with one change. Discrepancies between the number of connections arriving at each cell were removed, *i.e.* although destinations of connections were still random, each cell was assigned the same number of connections from cells within the same primitive and the same number of connections from cells in other primitives. Each cell had a total of 40 connections (twice the number shown in table 3.1 on page 67 as this network contained six primitives rather than three). If the destinations had been assigned randomly, on average one sixth of these connections would occur between cells in the same primitive, so in this experiment exactly 7 connections to each cell had to be connections from cells within the same primitive. The results of this experiment are shown in table 4.7. Clearly, success rates have plummeted, showing how important the lucky neuron effect is. Situation 1, on the other hand, in which one single primitive fails to ignite any others has actually improved. Removing the lucky neuron effect allows the generally inhibitory effect of the inter-primitive connections to predominate. The fact that correct ignition occurs in any of the situations may be attributed to the fact that cells in the externally stimulated primitives are not activated evenly. Once sufficient activity has been passed into a primitive, it is sustained by the relatively

| Situation tested | Success rate |
|---|---|
| 1 | 98.31% |
| 2 | 12.12% |
| 3 | 36.60% |
| 4 | 9.09% |

Table 4.7: Success rates for networks of six primitives with uniformly distributed connections.

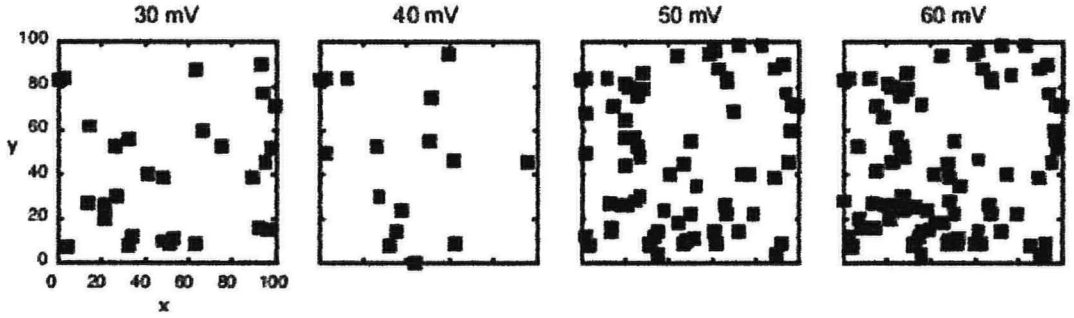


Figure 4.15: Development of lucky neurons in a cell assembly simulation by Igelsias *et al.* on a network of 10,000 cells (100 rows of 100 columns). Each black square represents a strongly connected cell. The number of such cells increases with higher external stimulation. Reproduced from [95].

high excitatory connections within that primitive. The success rate for situation 2 is somewhat misleading in the context of assessing the lucky neuron effect, since any situation in which the wrong combination of primitives ignites is counted as a failure. Removing this restriction (*i.e.* counting any combination of three active primitives as a success) virtually doubles the success rate to 22.81%.

The lucky neuron effect has been noticed in experiments by other researchers, although not named as such. Iglesias *et al.* [95] found that in a network of 10,000 cells in which a cell assembly was encouraged to form, approximately 8% of the excitatory cells formed strong connections with each other. This ratio is similar to the proportion of lucky neurons found in the experiments described here, although the number of strongly connected cells does increase with increasing external stimulus, measured by Iglesias *et al.* in terms of millivolts, as shown in figure 4.15.

Section 4.7 below describes an experiment in which weights were learned rather than determined by a genetic algorithm. The solution to the lucky neuron effect is not one that could have been derived from the learning procedure, since the weights were guided towards their final values by fixing the rate of co-firing between primitives. This is not to say that the lucky neuron effect does not take place in the networks with learned weights. Since the connection destinations are assigned randomly in both systems, it must do. It would appear that this effect is responsible for the relatively high level of erroneous activation of primitives and the corresponding low success rate when compared with the parameter set produced by the genetic algorithm. Apart from the weights between cells, the parameter set is the same for each system. The two parameter sets are therefore very close in the multi-dimensional parameter space. Had the learned weight values represented a better solution to the problem, the genetic algorithm would almost certainly have found it. Of course, it is perfectly possible for the genetic algorithm parameter set (or something very close to it) to be learned by adjusting the ratio in which the various training patterns were presented.

## 4.7 Learning weights in a hierarchy

This section describes an experiment in which weight strengths were learned in a hierarchy of cell assemblies in order to demonstrate that such learning is possible. A network of six primitives was created, each of 150 cells with the same parameter values as for previous experiments. However, weights were set to small random values, between 0 and 0.1 for excitatory weights, between 0 and −0.1 for inhibitory ones.

The network was subjected to a training regime in which relationships between primitives were established by co-activation. The following 2-3 cell assemblies were to be trained: $ABC$, $CDE$, $ADF$. Each primitive shares no more than one relationship with either of the others, so it should be possible to create three compound cell assemblies that can be ignited independently of the others simply by activating two of the constituent primitives. However, as shown in section 4.4 on page 88, conflicts can still occur. Primitive $A$ can still be ignited when $CDE$ was trained as it receives half its required activation energy from $C$ and half from $D$. It was found that this

problem could be avoided during training by ensuring that primitive $E$ was active for at least ten time steps before the external signal is applied to $C$ and $D$. As connections gradually approach their final values, the developing inhibitory connections from $E$ to $A$ discourage $A$ from igniting. It was found that this was sufficient to prevent erroneous ignition of $A$ in the great majority of cases.

Clearly it was necessary that stronger connections be established within primitives than between them, and this is easily arranged by frequency of co-firing. Each epoch of the training regime therefore consisted of one run of each of the training patterns shown in table 4.8 in a random order. Each run consisted of running the net for 300 time steps, with the appropriate primitives being externally activated for the first 10 steps. Learning was enabled throughout the entire training regime.

The number of times each training pattern was presented in order to ensure the correct co-activation frequency is also given. It was necessary to present some of the training patterns more often than others in order to maintain the correct ratio of co-firing of the neurons within and between primitives. This is necessary as it prevents primitives that are activated externally less often from being incorporated into more commonly activated primitives. Appropriate frequencies can be derived using the following argument:

1. It is necessary for excitatory connections within a primitive to be stronger than those between cells. Experiment 4.3 shows that the ratio should be about ten to one (0.5 as compared to 0.05). This is achieved by activating the individual primitives at least twice as often as the compound assemblies. It may appear that this would give a two to one ratio, but the reverberation effect, in which cells actually cause others to fire with the result that any connection between them strengthens, causes this ratio to rise. The strengths of the excitatory connections between cells in the same primitive do not rise indefinitely, however, as the external activation causes them to stabilise at or around 40%, as explained in section 1.1 (page 4).

2. A cursory glance at figure 4.8 shows that primitives $A$, $C$ and $D$ appear twice as often in the presentation of compound assemblies as do $B$, $E$ and $F$. It is

| times | $A$ | $B$ | $C$ | $D$ | $E$ | $F$ |
|---|---|---|---|---|---|---|
| 4 | × | √ | × | × | × | × |
| 4 | × | × | × | × | × | √ |
| 4 | × | × | × | × | √ | × |
| 2 | √ | × | × | × | × | × |
| 2 | × | × | √ | × | × | × |
| 2 | × | × | × | √ | × | × |
| 1 | √ | √ | √ | × | × | × |
| 1 | × | × | √ | √ | √ | × |
| 1 | √ | × | × | √ | × | √ |

Table 4.8: Training patterns with the number of times per epoch that each was presented. A tick indicates that a pattern is activated.

therefore necessary to present $B$, $E$ and $F$ twice as often as individual primitives as $A$, $C$ and $D$.

Combining these two precepts gives a desired ratio of two to one for patterns $B$, $E$ and $F$ compared to $A$, $C$ and $D$ and a ratio of (at least) two to one for patterns $A$, $C$ and $D$ compared to $ABC$, $CDE$ and $ADF$, *i.e.* the ratios shown in table 4.8.

Since the LTD used is post-not-pre (section 1.1 on page 4), the disparity of the number of runs in which single primitives were activated has a smaller effect on the connections between primitives than LTD using pre-not-post. For instance, imagine that the first training pattern were presented, in which cells in $B$ were activated. In the early stages of training, when weights are close to zero, connections from $B$ to other primitives are unlikely to be activated, so the cells in other primitives will not generally ignite. Post-not-pre LTD does not therefore reduce the connection strength, whereas pre-not-post LTD would do so, impeding the development of correct connections between primitives. Experiments were carried out in which pre-not-post LTD was used, and it was found that correct compound cell assemblies almost never formed.

The results of the learning experiment are shown in table 4.9. This table shows the mean weight value for each type of connection. Unlike networks whose parame-

| | *A* | *B* | *C* | *D* | *E* | *F* |
|---|---|---|---|---|---|---|
| *A* | 0.47 | 0.50 | 0.22 | 0.16 | 0.07 | 0.52 |
| | (-0.02) | (-0.07) | (-0.50) | (-0.61) | (-0.87) | (-0.10) |
| *B* | 0.38 | 0.51 | 0.44 | 0.16 | 0.11 | 0.09 |
| | (-0.13) | (-0.07) | (-0.11) | (-0.70) | (-0.67) | (-0.83) |
| *C* | 0.19 | 0.47 | 0.55 | 0.07 | 0.59 | 0.05 |
| | (-0.70) | (-0.09) | (-0.01) | (-0.55) | (-0.14) | (-0.93) |
| *D* | 0.12 | 0.13 | 0.09 | 0.48 | 0.77 | 0.81 |
| | (-0.66) | (-0.68) | (-0.44) | (-0.03) | (-0.12) | (-0.11) |
| *E* | 0.05 | 0.15 | 0.62 | 0.72 | 0.60 | 0.03 |
| | (-0.71) | (-0.82) | (-0.10) | (-0.11) | (-0.09) | (-0.07) |
| *F* | 0.74 | 0.03 | 0.13 | 0.59 | 0.02 | 0.48 |
| | (-0.19) | (-0.89) | (-0.88) | (-0.23) | (-0.57) | (-0.02) |

Table 4.9: Average weight strengths between cells in different primitives learned as a result of training. Inhibitory weights are shown in parentheses. The primitives down the left side represents the pre-synaptic primitives, the ones along the top the post-synaptic primitives.

| Situation tested | Success rate |
|---|---|
| 1 | 82.80% |
| 2 | 50.09% |
| 3 | 90.77% |
| 4 | 11.19% |

Table 4.10: Success rates for networks of six primitives for learned weights.

ters are determined entirely by genetic algorithms, each individual weight developed independently of the others. The stochastic nature of the initial weight strengths, their destinations and the input to the primitives results in a fairly widespread of final weight strengths but nevertheless a pattern does appear. Connections within primitives are strong. They have high excitatory weight strengths and inhibitory weights close to 0. Since there is a 40% probability that any post-synaptic cell has been externally activated given that the pre-synaptic cell is active, one might expect excitatory weights to converge on 0.4 and the inhibitory ones on 0.6. However, this ignores the fact that as connections strengthen, the pre-synaptic cell is more likely to excite the post-synaptic one and hence increase the likelihood of both cells being activated at the same time. The excitatory weight values learned did agree more closely with the weight value of 0.5 produced by the genetic algorithm. The proximity of these values is not coincidental, as the learned weight is constrained by the other parameters determined via the genetic algorithm.

Connections between cells in primitives with no relationship, such as $B$ and $E$, have low excitatory weights and inhibitory weights close to negative 1. Such cells should never fire at the same time. However, the connection strengths between cells in primitives that are occasionally co-active, such as $F$ and $D$, are also generally low with a wider variation in the inhibitory weights. Such connections undergo a mixture of LTP and LTD.

Experiment 4.4 on page 88 was repeated for the learned weights, the results being displayed in table 4.10. The most notable feature of table 4.10 is that success rates for the learned weights are substantially lower (except for situation 2) than for the parameter set determined by genetic algorithm, as shown in table 4.4 (page 90). This

is reasonable bearing in mind the fact that the genetic algorithm is programmed to find the parameters that give the best available performance, whereas the learned parameters simply reflect, to a greater or lesser degree, the co-firing rate of externally stimulated neurons. There is no reason to believe that such weights should be optimal in storing and retrieving cell assemblies. Indeed, it is perhaps surprising that the success rates for the learned parameters were as high as they were!

Success rates followed the same pattern for both learned and genetically determined weights, from which we may infer that the situations in which the network found itself played an important part in determining the result. For instance, the greatest success was found when two primitives not present in the same 2-3 assembly were activated. In the majority of cases, the activity in one or both of these primitives died out. The inhibitory connections between these primitives suggest that this should take place. In fact, activity in each primitive rests on a knife edge, teetering between extinction and uncontrolled activation, as explained in section 4.5 on page 94. Inhibitory connections between primitives has the effect of pushing each primitive activated towards extinction, and it was found that in the majority of the trials for situation 3, there was no activity in the network at all by time step 50.

## 4.8 Spontaneous Activation

There is some evidence that neurons in the brain occasionally fire without any external activation [2, 13], a phenomenon called *spontaneous activation*. It has been suggested that spontaneous firing of neurons in the visual cortex may be responsible for hallucinations [197, 50]. Hebb himself recognised the phenomenon [74, 76]. Neurons generally only fire spontaneously when they have been inactive for a long period (relative to their normal firing rate), typically 500 milliseconds. When a neuron has been active, it fatigues, which tends to discourage it from firing until it has recovered from that fatigue.

Why should neurons do this? It is conceivably possible that spontaneous activation is simply a biological occurrence, just as the human appendix has little purpose. However, any biological event usually has some cost incurred in its evolution, even if

only in terms of energy expended. There is some evidence to show that spontaneous activity is esponsible for segregating signals from the two eyes into occular dominance columns in the visual cortex [167, 114].

Hebb's theory suggests that activity in neurons is essential for cell assemblies to form, so spontaneous activation should have an effect on cell assembly formation and activation. Beurle [12] suggests that waves of activity passing through a large number of cells would use the presence of already activated cells that they encountered to maintain their activity level, change direction or even reflect on their original course.

Beurle's work also implies that spontaneous activity may allow more information to be stored in each wave. Spontaneous activation in territory which has not yet encountered the wave may cause cell in that region to fatigue, and become less responsive to activation by the wave when it arrives. In such a way, spontaneous activation prevents cell activity levels in the wave saturating, so that they can represent a wider variety of information signals.

For a connection to form between two cells in a cell assembly, they must both be active. While we can assume that some cells in any cell assembly are active at any one time, the same cannot be said of any uncommitted cells as there is no stimulus that would activate them. If uncommitted cells activated spontaneously at random intervals, then it is possible for connections between them and cells already recruited into cell assemblies to strengthen (as shown in figure 4.16(a)).

Without spontaneous activation it is still conceivably possible for a cell on the edge of a cell assembly to join that cell assembly provided it received enough activation to activate from a large number of cells via weak connections (as shown in figure 4.16(b)). However, activation by these means seems much less likely, as cells recruited in this manner would need several (weak) connections to cells already in the cell assembly, the only cells that would be guaranteed to activate at some point.

If spontaneous activation is in fact useful for enlargement of cell assemblies, the implication is that cell assemblies could expand more rapidly throughout the mass of cells available. The fact that cells to be recruited only need a few connections (possibly only one, in the extreme case) to the cell assembly allows it to "put out feelers" into the mass of uncommitted cells. The alternative, for cells to be recruited as in

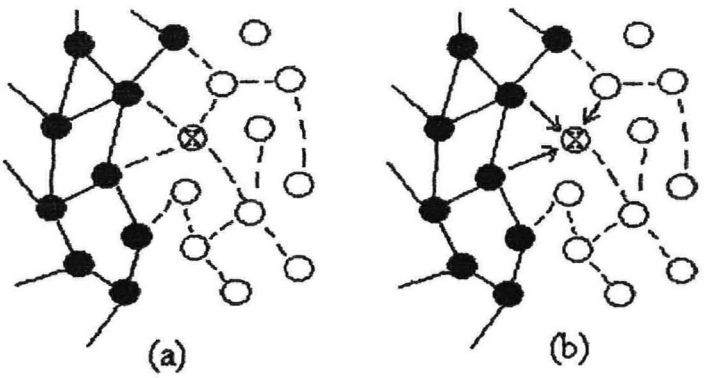


Figure 4.16: The black cells are committed to a cell assembly, the white cells are uncommitted. Solid lines represent strong connections, dashed lines and arrows weak ones. In (a), the cell marked X activates spontaneously, allowing it to be recruited into the cell assembly. In (b) X receives a small amount of activation from neighbouring cells as shown allowing it to activate.

figure 4.16(b), would restrict cell assemblies to growth only at their outer surfaces.

Simulations carried out by Huyck and Bowles [93] indicate that coupling spontaneous activation to a post-Hebbian learning rule keeps the weights of the neurons in a "state of readiness" so that they can easily be recruited into neighbouring cell assemblies. Reason suggests that a post-Hebbian rule in such a situation would be essential: the simple Hebbian learning rule, coupled with spontaneous activation of neurons which are far from any cell assembly would inevitably drive the weights of the uncommitted cells to zero, since the cells would only fire through spontaneous activation. As the simultaneous firing of two connected neurons would be a rare event, the primary force affecting the weights would be LTD, which tends to decrease the weights. Cells with weights close to zero would be unlikely recruits to any cell assembly that might expand into their "territory."

Experiment 4.8.1, described below, investigates the effects that differing levels of spontaneous activation have on the formation of cell assemblies in a grid of cells. This may, in turn, throw some light on whether spontaneous activity aids formation of cell assemblies in the human brain.

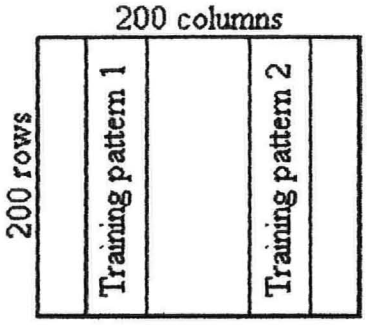


Figure 4.17: Training patterns in the spreading activation experiment.

### 4.8.1 Spread of Cell Assemblies through spontaneous activation

An experiment was carried out to investigate whether cell assemblies would recruit unaffiliated cells at their edges. A grid of cells, 200 rows by 200 columns, was set up. The grid was presented with two input patterns, random activation of cells in a rectangular band covering the whole of columns 30 to 70, and random activation of cells in a rectangular band covering the whole of columns 130 to 170, as shown in figure 4.17. In addition to these input patterns, each cell had a given probability during training only of activating spontaneously when inactive.

After training, it was found that the grid of cells recognised the two training patterns. Partial activation of a training pattern would lead to almost complete activation of that pattern. Figure 4.18 shows a cross section of the grid along the middle row.

In figure 4.18, the solid lines show the extent of the cell assemblies when there is no spontaneous activation. It can be seen that without spontaneous activation, the cell assemblies do not spread at all beyond the boundaries of the input patterns themselves. However, the experiment also shows that cell assemblies do in fact recruit cells that receive no input other than spontaneous activation. The dotted lines represent the extent of the cell assemblies when each inactive cell had a probability of 5% of activation, and the dashed lines represent the same with a probability of 10%. Both of these extend beyond the confines of the two input patterns. As the probability of spontaneous activation increases, so the likelihood of the cell assemblies

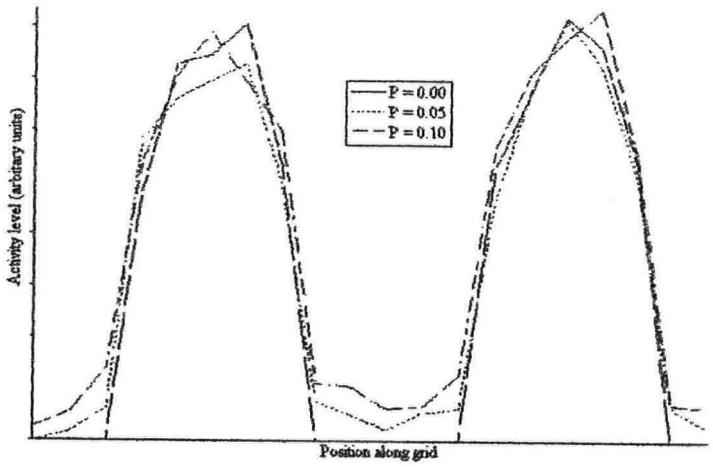


Figure 4.18: Recruitment of cells through spontaneous activation. A cross section through the grid shows the number of cells that activate.

spreading into adjacent territory increases. Experiments showed that increasing the probability above about 25% led to each test pattern activating both cell assemblies. Effectively, both cell assemblies had merged into one large one due to connections forming across the gap between the two input patterns.

### 4.8.2 Cell assembly ignition from spontaneous activation

It is perfectly reasonable to suppose that spontaneous activity itself could be responsible for activating a previously trained cell assembly. Often, ideas will simply "pop into one's head" for no apparent reason, and spontaneous activation of a cell assembly may be the cause of this.

A further experiment was carried out to analyse the degree to which spontaneous activation of cells could activate a cell assembly. A grid of cells, 20 rows by 20 columns, each with a connectivity to 6 others assigned randomly, was created and a cell assembly was trained in the central 100 cells by randomly stimulating each cell with a probability of 0.2 over a number of time steps. It was found that no more than 50 time steps were required for cell assemblies to form reliably in 97% of trial runs. No spontaneous activation of the cells outside the central square was permitted, although cells in that region could be activated by their connections. The topology

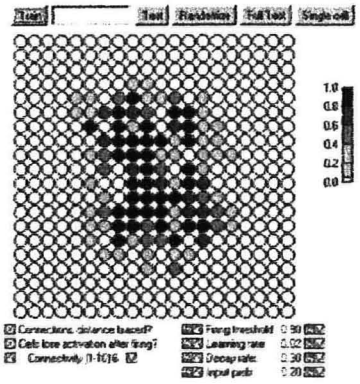

Figure 4.19: Topology of a 20-by-20 network subject to spontaneous activation, showing a trained cell assembly. The cell assembly has developed in the central 10-by-10 square of cells, and has succeeded in passing some activation to cells on its borders, although none of the cells outside the central square has actually fired (activity level greater than the firing threshold of 0.9).

of the network, together with a trained cell assembly, is shown in figure 4.19.

After training, the grid was run for several time steps, taking spontaneous activation of the cells as its only input. The number of time steps required for the cell assembly to activate was noted, where activation was defined as being when at least 70% of the cells that had been active directly after training were active. The average number of time steps required for the cell assembly to be activated is shown in figure 4.20.

It was found that spontaneous activation was not always enough to activate the trained cell assembly. Figure 4.21 shows the proportion of the test runs where the cell assembly was triggered for differing levels of spontaneous activation.

Repeated experiments showed that the length of time required for the cell assembly to ignite, and the probability of that ignition, depended strongly on the level of spontaneous activation of the cells. Unsurprisingly, the greater the probability of activation, the greater the proportion of test runs where ignition took place, and the sooner that ignition happened. With activation probability greater than about 0.01, cell assembly activation was virtually guaranteed.

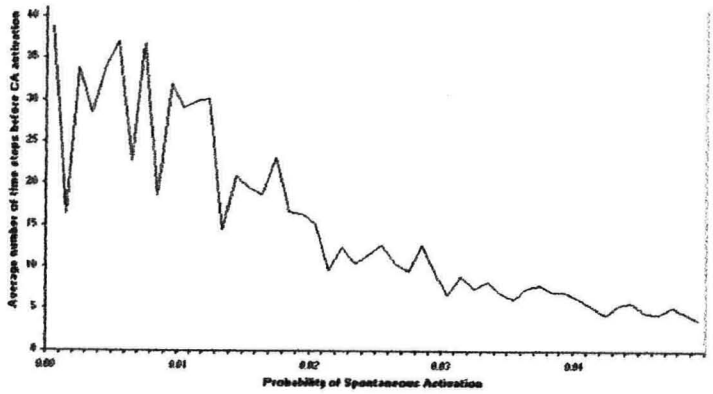


Figure 4.20: Number of time steps required for cell assembly activation.

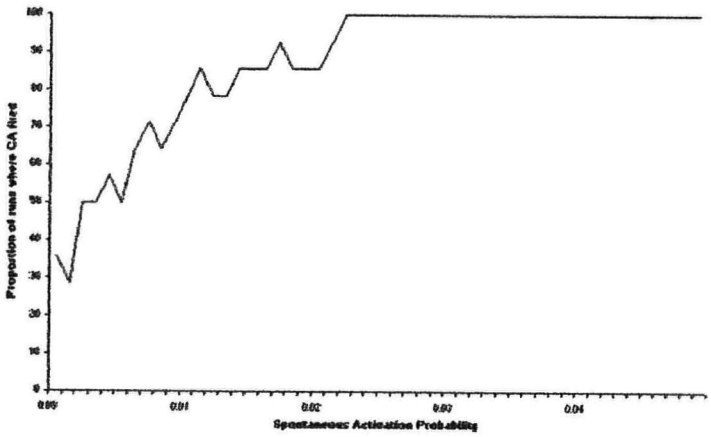


Figure 4.21: Frequency of spontaneous cell assembly ignition.

### 4.8.3 Forgetting of cell assemblies

It is said that one never forgets how to ride a bicycle, even if one has not mounted one for years. If there is any validity in the theory of cell assemblies as devices in the human brain which store memories, then we must assume that the skills required to ride a bicycle are stored in one or (almost certainly) more cell assemblies, and that these persist in the brain for many years, whether the owner of that brain gets on a bicycle or not.

The experiments above have shown that spontaneous activation is responsible for the augmentation and change of cell assemblies. In this respect, spontaneous activation represents a hazard to rarely-used cell assemblies such as the bike-riding cell assembly. If the cell assembly is rarely activated, then spontaneous activity of cells within it would encourage them to be recruited into neighbouring cell assemblies with different purposes, and we would forget how to ride a bicycle. Since this does not appear to be the case, not only with bike-riding but with a large number of rarely used skills and known facts, there must be some mechanism that prevents this from happening.

Paradoxically, spontaneous activation may come to our aid here. It may be possible for some cells in a cell assembly to be activated without the whole activating, or possibly for one cell assembly involved in the skill of bike-riding to activate without activating them all. Such cell assemblies would be *subassemblies* to the bike-riding cell assembly. This would have the effect of strengthening the connections between the cells, but would not activate the entire arrangement of neurons associated with bike-riding, thereby preventing the thought of riding a bike from "breaking the surface" of our consciousness. If spontaneous activation of the bike-riding cell assembly, or series of cell assemblies, always resulted in full activation, then the idea of riding a bicycle would occur to us regularly (every few minutes or seconds) without fail, and this is clearly not the case either.

I have already shown that spontaneous activation can cause a cell assembly to ignite. The following experiment was designed to show whether it can cause the connections in a cell assembly to weaken to the point where the cell assembly is effectively disbanded, in other words, whether spontaneous activation can cause a

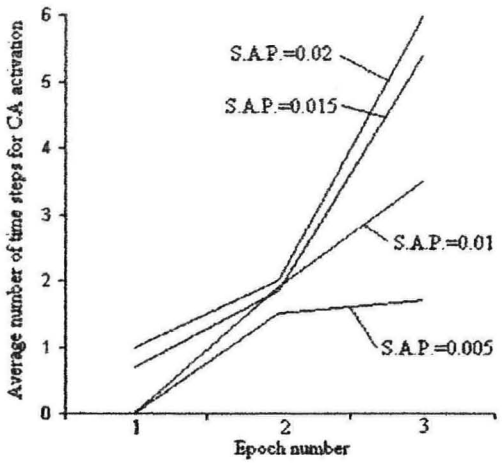


Figure 4.22: Forgetting of cell assemblies after 1, 2 and 3 epochs of 50 time steps with solely spontaneous activation. S.A.P. = spontaneous activation probability. There was no forgetting when S.A.P. = 0.

cell assembly to forget.

A grid of cells was set up with an identical topology and training regime to the previous experiment. It was tested in a slightly different manner, however. This time connections were adjusted throughout testing in the same manner to that throughout training. Every 50 time steps (termed an "epoch"), the connections were "frozen" and a test pattern that had previously been found to activate the cell assembly was tested to see if it still activated the cell assembly. This was repeated for as many epochs as was required for the test pattern to fail to activate the cell assembly.

Figure 4.22 shows the average number of time steps required to activate the cell assembly after 1, 2 and 3 epochs for different levels of spontaneous activation. Spontaneous activation probability was limited to values less than 0.05 as any probability above that caused unwanted activation of the cell assembly during the forgetting period between tests, thereby strengthening the connections. It was found that probabilities between 0.02 and 0.05 caused the cell assembly to be forgotten immediately (*i.e.* it was not even activated after one epoch).

This experiment indicates rather predictably that increased levels of spontaneous activation probability leads to rapid forgetting of cell assemblies. It becomes ever harder to activate cell assemblies and after approximately 4 or 5 epochs (regardless of

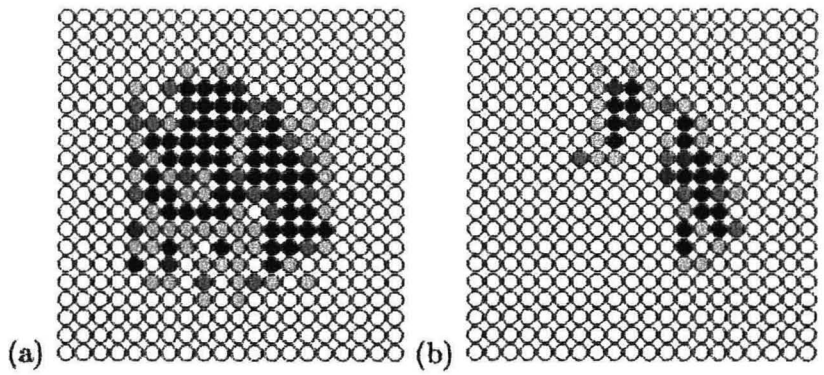


Figure 4.23: Spontaneous fractionation of cell assemblies. (a) The cell assembly directly after training. (b) The cell assembly has split into two sections, each of which can be reliably ignited without the other igniting.

the probability level) the cell assemblies decay to the point where they are impossible to activate. It was found that if the probability was increased to the point where the cell assembly was activated purely by spontaneous activation, the problem of forgetting disappeared entirely. Even after 20 epochs, the cell assembly was reliably activated and for some high levels of probability, its size increased with time due to recruitment.

### 4.8.4 Fractionation of cell assemblies

One common side effect that was noticed as increased levels of spontaneous activation were introduced in the last experiment was that cell assemblies tended to fractionate into smaller cell assemblies, each of which was self-sustaining and could be ignited independently of the other. Whenever this happened, that particular test run was not included in the results for experiment 4.8.3 on page 113. Figure 4.23 shows an example of a cell assembly that fractionated after 3 epochs at spontaneous activation probability 0.02.

Fractionation of cell assemblies has important ramifications for Hebb's theory of human memory. The ability of a cell assembly to fractionate allows it to *specialise*. For instance, in infancy, children learn to classify all four-legged animals that they encounter as "horse" (or sometimes "dog") [18]. Only later, do they learn to differ-

entiate between different domestic animals. Similarly, when learning the past tenses of verbs, children initially learn that the past tense is formed by adding "-ed" to the verb stem: worked, learned etc. They cannot cope with strong verbs such as "come," often turning them into "comed" or "camed." Later, they learn the correct past tense of these verbs [152].

Hebb's theory would suggest that this increasing specialisation of knowledge comes from cell assemblies splitting into smaller units. Initially, a cell assembly develops which covers the entire general concept ("horse" or "-ed"). Gradually, the inputs fall into different categories (for instance, large domestic animals as opposed to small ones), and the cell assembly fractionates into two smaller sections. The large domestic animals are classified as "horse," and the small ones as "dog." Later still, these specialist cell assemblies fractionate, as the children learn to differentiate different breeds of dog. We may postulate the setting up of a hierarchy of cell assemblies in which knowledge becomes stored. We all recognise that different breeds of dog are still members of the primitive "dog" and that dogs are members of the primitive "four-legged animals."

Fractionation of cell assemblies to represent ever more specialised concepts is an example of what Sakurai [154] terms population ensemble coding. He concludes that neurons must co-operate in order to represent knowledge and memories. He gives several reasons why this must be the case:

1. There is a virtually unlimited number of information items that need to be stored in the human memory. Not only are there single items, such as "dog" or "cat", but these can be joined with adjectives and the like to form an unlimited number of combinations. Even if every neuron in the human brain were dedicated to storing knowledge, there would not be enough neurons to encode all the possible information that humans have to deal with. Yet, humans do manage to store all these information items.

2. It is inconvenient to store the similarities between items using totally separate neuronal codings. For example, Dylan the golden retriever and Merlin the golden retriever share many items in common, which must be represented in

the brain of their owner in the form of neuronal codings. It would be highly inefficient to represent all the information known about the two dogs using totally separate codings, which implies that the two codings share a lot of the neurons in common (indeed in common with all the other dogs that the owner knows).

I am not suggesting that spontaneous activity in the brain is the root cause of cell assembly fractionation. However, since fractionation occurs in approximately 20% of the cell assemblies formed in the last experiment, spontaneous activity does seem to be a factor that is likely to cause fractionation, and this phenomenon is worth further investigation.

### 4.8.5 Discussion

These experiments have been designed to show that spontaneous activation of cells during training of cell assemblies is both desirable and useful. It leads to more robust cell assemblies being created, and to increased likelihood of cell assemblies being activated, which in turn leads to the concepts represented by the cell assemblies being "rehearsed," which may be an essential part of keeping memories that are rarely consciously called upon. It offers an interesting line of research into how cell assemblies fractionate to become more specialised.

Human memory is a strange phenomenon. Actions which are repeated often are understandably generally remembered. Indeed, this is the rationale behind such activities as practising a musical instrument or learning to touch type. However, memories can be retained indefinitely on the basis of events that happen only once, such as traumatic experiences like bereavement or violence. Memories from childhood are often retained over much more recent ones. Indeed, memories of events that have not even taken place can be retained for years. There is anecdotal evidence of people being able to remember particularly vivid dreams or nightmares, people having apparent memories implanted during hypnosis. The author Marcel Proust, in his book *A la Recherche du Temps Perdu* vividly describes an incident that he remembered from his infancy of being kidnapped from his pram, which later turned out to be nothing more than a fantasy invented by his nurse.

Since what is remembered and what is not seems subject to some certain lottery, it is reasonable to assume that a complex interplay of cell assemblies should be involved in the formation, retention and adaptation of memories, and that spontaneous activation should play some part in that. The stochastic nature of spontaneous activation may go some way toward explaining the unpredictable nature of human memory, and why it cannot be relied upon to give us a true account of what we have experienced and what we have not.

## 4.9 Summary

In this chapter I have shown how increasingly complicated networks of cell assemblies can be created in a network of identical cells simply by predetermining the connections between them. These cell assemblies form a simple hierarchy, with primitive cell assemblies, self-contained primitives of cells that are capable of sustaining activation, in turn activating primitives of cell assemblies. The exact behaviour of the cell assemblies depends on a series of parameters that must take tightly constrained values. These parameters co-operate in a complicated fashion that makes predicting the exact behaviour of the cells *en masse* very difficult, a problem that is compounded by the fact that destinations of connections between cells are assigned randomly. Although the networks do not always behave in the way one would first expect, it is usually possible to come up with an explanation afterwards that accounts for the behaviour. Dissection of the networks and further experiments can then be used to justify or nullify the explanation. This is indeed true of most of the experiments described in this thesis.

Although the parameters control the behaviour of the network in a way that is difficult to predict, it is still possible to draw some general conclusions about them. Three of the parameters appear to be mainly responsible for the transmission of activation energy between one primitive cell assembly and another: the excitatory and inhibitory connection strength between cells of different primitive cell assemblies, and the number of connections per cell. The other parameters appear mainly to control activity within primitive cell assemblies. Ignition of primitives is a combination of

external activity and retained activity. A low retention rate, for instance, undermines a primitive's ability to resist fatigue in the absence of external stimulation. Even a relatively high external activation may not allow the primitive to ignite if many of the cells have already fatigued. The interplay of the different parameters is the main reason why there are few detailed mathematical conclusions in this chapter.

I show how a simple Hebbian training rule allows connection strengths to develop between cells according to the frequency of activation. This procedure works both in individual cell assemblies and in simple hierarchies of cell assemblies, although the resulting weight values do not match the theoretical values due to the complex interactions between cells. However, I have demonstrated that such learning is sufficient to learn simple relationships between primitives. I have also shown that cell assemblies can manifest other behaviours such as recruitment and fractionation through the effect of spontaneous activation of cells. Clearly networks of primitives can both form hierarchies, and subdivide in order to store knowledge at different levels of refinement, thereby showing a great deal of potential for adaptation during use. I have shown through experiments that this potential arises from the relatively simple mechanisms of Hebbian learning and spontaneous activation. This is a possible model for how cell assemblies develop and evolve in the brain, although it is by no means certain.

A comparison between the performances of the parameter set involving learned weights with that involving evolved weights produced an interesting and unexpected conclusion. The genetic algorithm resulted in counter-intuitive weight settings, through which the total amount of activation passed from one primitive to another was in fact negative. While this may appear to eliminate the possibility of any primitive igniting another, it does in fact act as a control on what would otherwise be uncontrolled ignition. A small proportion of the cells, that happen to be well connected, provide a large proportion of the activity, easily enough to ignite a primitive, and that the negative connections are necessary to rein back this activity so that the primitive only ignites when provided with energy from two primitives. I term this effect the "lucky neuron" effect. Indeed, the solution produced by the genetic algorithm gave a higher success rate than the learned weights, showing that the overall inhibitory effect of the weights can restrain the lucky neuron effect and regulate the ignition of

primitives.

As the number of primitive cell assemblies increases, the possible complexity of the hierarchies also increases. More arrangements of primitive cell assemblies become possible. This can lead to situations in which activating two primitives sets off a convoluted chain of ignition and suppression, so care has to be taken when assigning compound assemblies to be included in the network.

In general, parameters controlling behaviour need not change as the number of primitive cell assemblies increases, the one exception being the number of connections between cells. Since increasing the number of primitive cell assemblies means greatly increasing the number of cells in the network, the number of connections per cell must be scaled up accordingly in order for the connection density of the cells to remain approximately constant.

I have succeeded in showing that a network containing as few as six primitives is capable of a variety of behaviours, some very unexpected. In chapter 5 on page 121, I extend the size of the networks indefinitely and show how the number of possible cell assemblies increases with network size. The scope for unexpected results also increases a great deal, so the topologies of the networks are strictly limited to those which minimise the possibility of unexpected patterns of ignition. Chapter 5 also sees experiments on large scale Hopfield nets, using similar topologies to the cell assembly networks.

# Chapter 5

# The capacity of an associative memory

The experiments described in chapter 4 on page 72 showed that primitive cell assemblies can be combined to produce a compound cell assembly which ignites if a sufficient number of its component primitives are ignited. The chapter also showed that adding more primitives to an existing network of cells allows further compound cell assemblies to be created, but that compound cell assemblies can be ignited erroneously.

Three network topologies are considered in this chapter. The first topology, defined in section 5.1 on page 122, is investigated in an experiment to show that a network of cells can only hold $O(n)$ primitives if each cell has a fixed number of connections regardless of the number of cells in the network, where $n$ is the number of cells in the network. In order for $O(n)$ primitives to be stored, the destinations of the connections must be partially directed rather than assigned completely at random. The second is a network that consists entirely of 2-3 cell assemblies. It is shown that a network of $N$ such primitives gives a storage capacity on the order of $N$. The third topology is that described in [94] which states that a network consisting of $N$ 3-4 cell assemblies can, in principle, store on the order of $N^2$ cell assemblies. I show that this model contains several presumptions that may not be valid in a biological system.

Experiment 4.4 on page 88 has shown that the combining of primitives to form compound cell assemblies is a complicated matter. Although primitive cell assemblies

can be combined recklessly to give any number of different compound cell assemblies, care has to be taken to ensure that erroneous combinations are not included. For instance, given primitive cell assemblies termed $A$ to $F$, combinations $ABC$, $CDE$, $ADF$ and $BEF$ can theoretically be included. However, as has been seen in the previous chapter, unintended combinations $ACD$ and $BEC$ are also created. This occurs because each primitive in a 2-3 cell assembly requires half its activation energy to come from each of the other two members in order to be activated. Activating $A$ and $D$ in order to activate $F$ (in the combination $ADF$) also provides the required energy to $C$, half from $A$ in $ABC$ and half from $D$ in $CDE$. In practice, $C$ and $F$ compete for activation, the winner shutting the other one down.

## 5.1 Networks of cells with fixed numbers of connections can store $O(n)$ primitives

Experiment 4.1.1 on page 75 shows that it is possible to establish a single cell assembly in a grid of cells. To create a large-scale network of cells containing a number of primitives that is proportional to the number of cells is a trivial matter: Just duplicate the architecture in experiment 4.1.1 as many times as needed, with no overlapping connections from one primitive to the next. In such an architecture, the number of connections from each cell is limited to 6, the number preferred by the genetic algorithm to maintain activity in the single cell assembly. Assuming that the connections within each primitive are assigned randomly, the distributions of cells near the edge of each primitive do not follow the same pattern as those of cells closer to the centre of the primitive, as shown in figure 5.1.

Experiment 5.1 was carried out on a network of cells with partially targeted connections, in which connections from cells were assigned randomly within their own primitives. However, a certain degree of overlap was allowed between primitives, as shown in figure 5.2, so that the cells at the boundaries of the primitives were members of both. Admittedly, this violates the stated principle that primitives should contain no cells that were members of other primitives, but the intention was not that these cells should in themselves be able to ignite either primitive, rather that the degree

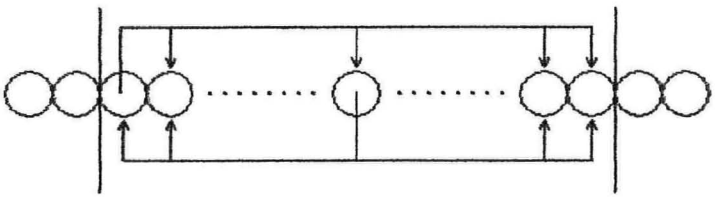

Figure 5.1: Inserting "watertight" barriers (between primitives causes distributions of connections from cells towards the edge of each primitive to be skewed towards the centre, whereas distributions of connections from cells nearer the centre are more symmetrical. (The arrows show possible destinations from each of the two cells marked.)

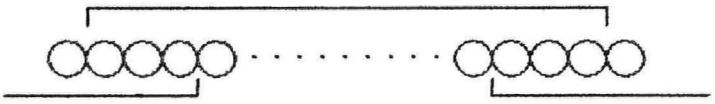

Figure 5.2: Allowing a certain degree of overlapping between primitives allows for more compact storage of primitives in a network of a given size.

of overlap should be small enough so that it should be possible to activate each primitive independently of the other. Including such an overlap allows for more compact storage of primitives, i.e. $n < Ng$ where $g$ is the number of primitives. The experiment investigated how the ability to activate each primitive individually changed as the degree of the overlap increased. The parameter values used were the same as for experiment 4.1.1, with the number of primitives being set at 20. Connection strengths were predetermined rather than learned. Primitives at the extreme ends of the network were allowed to overlap in a "round the world" fashion.

For each degree of overlap, each primitive was activated in turn 100 times by activating each cell with a probability of 0.4 as in experiment 4.1.1. Each trial had a different set of connections, assigned randomly but adhering to the restrictions given. Table 5.1 shows the percentage of all the activations that did not lead to any neighbouring primitive igniting (10 or more cells of the primitive firing at any given time step).

Clearly, with no overlap between primitives there is no possibility of erroneous

| Degree of overlap (cells) | Percentage success |
|---|---|
| 0 | 100 % |
| 1 | 100 % |
| 2 | 99.2 % |
| 3 | 90.6 % |
| 4 | 81.4 % |

Table 5.1: Increasing overlap of primitives results in increasing degrees of erroneous ignition.

ignition of primitives. This proves that a network of $n$ cells can support $O(n)$ independent primitives. However, as the degree of overlap increases, the probability of neighbouring primitives being ignited increases greatly. Can a network of $n$ cells, each with a fixed number of connections, hold $O(n^2)$ cell assemblies, primitive or compound? A later experiment, 5.2.2(c), shows that fixing the number of connections reduces the probability that compound assemblies can be ignited reliably as the size of the network increases.

## 5.2 2-3 Cell Assemblies have a storage capacity of $O(n)$

This section describes work carried out on large networks of primitives which are connected to form 2-3 cell assemblies, *i.e.* forming compound assemblies containing three primitives. Activating any two of these is sufficient to ignite the third, but activating a single primitive is insufficient to ignite either of the other two. Experiments described in chapter 4 on page 72 have shown that it is possible to create such a compound assembly in a network containing only three primitives. It would be a trivial matter to create any number of such compound assemblies in a large network providing that connections between primitives did not cross from one compound assembly to another. The experiments described in this section demonstrate that it is possible for compound assemblies to share primitives.

ABC EFG IJK MNO QRS UVW YZa ... ...
CDE GHI KLM OPQ STU WXY abc ... ...

Figure 5.3: Triplets of primitives

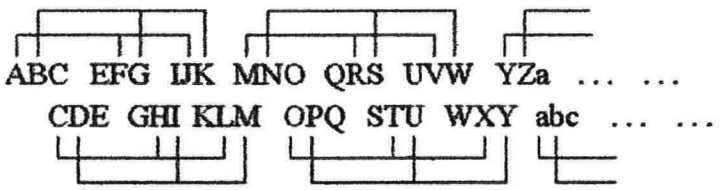


Figure 5.4: Solid lines indicate further compound cell assemblies.

### 5.2.1 Theoretical considerations

In the following description, all primitive cell assemblies are referred to using upper and lower case letters, with $A$ referring to a different primitive to $a$ etc. It is convenient to split the compound cell assemblies constructed from these primitives into various levels, each of which may be considered separately.

Firstly, primitives may be grouped into triplets, as shown in figure 5.3. Two primitives in each triplet take part in more than one compound assembly. Given $N$ primitives, this gives approximately $\frac{N}{2}$ compound cell assemblies.

Further compound cell assemblies can be formed via strong connections between primitives in different triplets. Clearly, no such connection should be made if it leads to conflict. For instance, no connection should be formed between $A$ and $E$ as they both already share a connection to $C$. However, $A$ can be connected to $F$ without any such conflict. Adding a layer of connections as shown in figure 5.4 gives approximately $\frac{N}{3}$ further compound cell assemblies.

Further layers can be added by connecting primitives in a similar way to that of figure 5.5. The next layer of connections involves primitives being connected to another in the next triplet but two (*i.e.* connections "skip over" triplets), as shown in figure 5.5. This gives approximately $\frac{N}{6}$ further compounds cell assemblies. At each layer of the hierarchy, care must be taken to avoid conflicts with already existing compounds. For instance, in the layer above the one shown in figure 5.5,

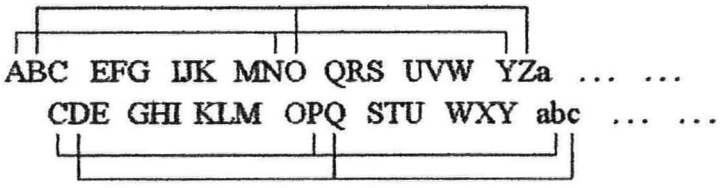


Figure 5.5: Solid lines indicate further compound cell assemblies.

primitives within $ABC$ are connected to those in $cde$ rather than the more obvious choice of $QRS$, as connections exist between primitives in $ABC$ and those in $QRS$ via primitives in $MNO$. The component primitives of each compound assembly are determined simply by connecting the first primitive ($A$) to the next two such that there would be no conflict with any previously assigned compound assembly. Once such a triplet of assemblies has been found, the process is then repeated starting with the next primitive after the last one in the triplet. A computer program was then used to confirm that no conflicts were present given any assignment of connections.

Each of these layers provides the order of $N$ compound cell assemblies, so the total number of compound cell assemblies must also be of the order of $N$. The number of layers that can be added depends on the value of $N$. For instance, layer 3 can only be added once the number of primitives has reached 27. At first sight, it might appear that the actual capacity is therefore of the order of $N^2$. However, the "span" of each compound cell assembly across the list of triplets increases as further layers are added. In addition, compound cell assemblies within layers cannot be allowed to overlap to any great extent, or conflicts would occur. For this reason, I maintain that the storage capacity cannot be said to be of the order $N^2$.

Assigning each primitive a number ($A = 1$, $B = 2$, $a = 27$ etc.) allows the relationships to be determined mathematically. For the triplets at level 1, cell assemblies are determined by the pattern $x$, $x+1$, $x+2$ for all $x = 2n+1$, $n = 0, 1, 2, \ldots$ At level 2, the pattern is $x+y$, $x+y+5$, $x+y+9$ for all $x = 12n+1$, and $y = 0, 1, 2$ or 3. At level 3, this becomes $x+y$, $x+y+13$, $x+y+25$ for all $x = 28n+1$, and at level 4, the pattern is $x+y$, $x+y+57$, $x+y+103$, for all $x = 106n+1$. For an infinite number of primitives there is no limit to the number of levels, although I have not, as yet, been able to produce a single equation that covers all levels.

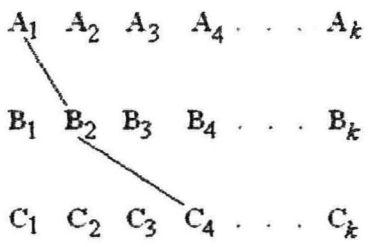


Figure 5.6: Creating 2-3 cell assemblies within a network arranged into three groups of primitives.

It is still necessary to have strong inhibitory connections between primitives that are not present in any compound assembly. For instance, if $A$, $B$, $G$ and $H$ were all active simultaneously, $C$ and $I$ would ignite. This would lead to the erroneous ignition of $D$. Strong inhibitory connections between $A$ and $G$, and between $B$ and $H$ would help to prevent this.

Any rigorous attempt to store $N^2$ 2-3 cell assemblies in a network of primitives leads to failure. Consider the situation in which primitives are arranged into three groups $(A_1 \ldots A_k, B_1 \ldots B_k, C_1 \ldots C_k)$, as shown in figure 5.6. Weights may be set so that 2-3 cell assemblies consist of one primitive from each group. For example, figure 5.6 shows the cell assembly $A_1B_2C_4$. To store $N^2$ cell assemblies, each cell in at least two of the letters must be involved in $O(N)$ cell assemblies. Figure 5.7 shows one way to achieve this, which, superficially at least, appears to be satisfactory.

However, even a cursory analysis of such a network reveals that it is impossible to set up on the order of $N^2$ compound cell assemblies in it. For this to be the case, each of the $A$ primitives (for example) would have to be connected to each of the $B$ primitives (for example), with the $C$ primitives arranged so as to avoid conflicts. A possible arrangement is shown in figure 5.7.

Consider the situation in which $A_1B_1C_1$ is ignited by activating $A_1$ and $B_1$. $C_1$ receives half the activation energy it needs to ignite from each of $A_1$ and $B_1$ and duly ignites. However, $C_2$ also receives half the activation energy it needs to ignite from $A_1$ thanks to $A_1B_2C_2$, and the other half from $B_1$ thanks to $A_2B_1C_2$. Once $C_2$ has ignited, further 2-3 cell assemblies ignite (such as $A_2B_1C_2$) and activity spreads rapidly throughout the network, as shown in figure 5.8. Other arrangements give

| | | | | | | |
|---|---|---|---|---|---|---|
| $A_1B_1C_1$ | $A_2B_1C_2$ | $A_3B_1C_3$ | ... | $A_{k-2}B_1C_{k-2}$ | $A_{k-1}B_1C_{k-1}$ | $A_kB_1C_k$ |
| $A_1B_2C_2$ | $A_2B_2C_3$ | $A_3B_2C_4$ | ... | $A_{k-2}B_2C_{k-1}$ | $A_{k-1}B_2C_k$ | |
| $A_1B_3C_3$ | $A_2B_3C_4$ | $A_3B_3C_5$ | ... | $A_{k-2}B_3C_k$ | | |
| $\vdots$ | $\vdots$ | $\vdots$ | | | | |
| $A_1B_{k-2}C_{k-2}$ | $A_2B_{k-2}C_{k-1}$ | $A_3B_{k-2}C_k$ | | | | |
| $A_1B_{k-1}C_{k-1}$ | $A_2B_{k-1}C_k$ | | | | | |
| $A_1B_kC_k$ | | | | | | |

Figure 5.7: One possible arrangement of primitives to form 2-3 cell assemblies.

slightly better performance in which the network eventually achieves a stable state with approximately 20% of the compound cell assemblies having ignited.

More generally, consider cell assemblies $A_pB_yC_r$, $A_pB_qC_z$ and $A_xB_yC_z$, as shown in figure 5.9. Igniting $A_pB_yC_r$ by activating $A_p$ and $B_y$ will always ignite $C_z$ erroneously as well as $C_r$ due to $C_z$ being present in each of the two other compound cell assemblies. Although figure 5.7 shows a fairly simple arrangement of $B$ and $C$ indices, however complex the patterns of the indices used, it is impossible to produce an arrangement in which this problem does not occur and which still stores $O(N^2)$ patterns. Conflict can be avoided by reducing the number of $A$ primitives involved in compound cell assemblies (e.g. reducing the table in figure 5.7 to two or three columns), but this would reduce the number of patterns stored to $O(N)$.

### 5.2.2 Experimental results

Experiment 5.2.2(a) investigated how many 2-3 cell assemblies could be stored and retrieved reliably in a network of a given number of primitive cell assemblies. A network of 150 cells per primitive was set up for a variable number of primitives. The parameter values for the cells were the same as those listed in table 3.1 on page 67, except that the number of connections from each cell were to be scaled up according to network size.

The purpose of the experiment was to determine the optimum number of connections per cell and the optimum connection strength between primitives. As demonstrated in chapter 4 on page 72, the activation energy passed between primitives

1. Consider a network arranged as in figure 5.7 with $k = 5$. When $A_1$ and $B_1$ are activated, $C_1$ activates immediately (active primitives are shown in upper case, inactive ones in lower case).

   $A_1B_1C_1$ $a_2B_1c_2$ $a_3B_1c_3$ $a_4B_1c_4$ $a_5B_1c_5$
   $A_1b_2c_2$ $a_2b_2c_3$ $a_3b_2c_4$ $a_4b_2c_5$
   $A_1b_3c_3$ $a_2b_3c_4$ $a_3b_3c_5$
   $A_1b_4c_4$ $a_2b_4c_5$
   $A_1b_5c_5$

2. Primitives $A_2$, $C_2$, $A_3$, $C_3$, $A_4$, $C_4$, $A_5$ and $C_5$ receive half the necessary activation from $B_1$. $B_2$, $C_2$, $B_3$, $C_3$, $B_4$, $C_4$, $B_5$ and $C_5$ receive half the necessary activation from $A_1$. As a result, all $C$ inactive primitives ignite.

   $A_1B_1C_1$ $a_2B_1C_2$ $a_3B_1C_3$ $a_4B_1C_4$ $a_5B_1C_5$
   $A_1b_2C_2$ $a_2b_2C_3$ $a_3b_2C_4$ $a_4b_2C_5$
   $A_1b_3C_3$ $a_2b_3C_4$ $a_3b_3C_5$
   $A_1b_4C_4$ $a_2b_4C_5$
   $A_1b_5C_5$

3. All compound cell assemblies in the first row and first column now ignite.

   $A_1B_1C_1$ $A_2B_1C_2$ $A_3B_1C_3$ $A_4B_1C_4$ $A_5B_1C_5$
   $A_1B_2C_2$ $A_2b_2C_3$ $A_3b_2C_4$ $A_4b_2C_5$
   $A_1B_3C_3$ $A_2b_3C_4$ $A_3b_3C_5$
   $A_1B_4C_4$ $A_2b_4C_5$
   $A_1B_5C_5$

4. Now all compound cell assemblies are either complete or have two primitives active. Inevitably, all cell assemblies in the network now ignite.

   $A_1B_1C_1$ $A_2B_1C_2$ $A_3B_1C_3$ $A_4B_1C_4$ $A_5B_1C_5$
   $A_1B_2C_2$ $A_2B_2C_3$ $A_3B_2C_4$ $A_4B_2C_5$
   $A_1B_3C_3$ $A_2B_3C_4$ $A_3B_3C_5$
   $A_1B_4C_4$ $A_2B_4C_5$
   $A_1B_5C_5$

Figure 5.8: Runaway ignition among primitives

$A_pB_yC_r \quad \ldots \quad A_xB_yC_z$

$\vdots$

$A_pB_qC_z$

Figure 5.9: Erroneous activation of $C_z$

does not bear a simple relationship to the probability of igniting destination cell assemblies. Instead, ignition appears to depend on a sufficient number of cells in the destination cell assembly being connected in such a way that they receive more activation from excitatory connections than they lose via inhibitory ones. A low weight strength for excitatory nodes (much less than 0.2) makes this statistically unlikely. A large number of connections between cells also reduces the likelihood of cells in the destination primitive receiving a net positive energy as the average energy passed to any cell becomes more representative of the excitatory weight value. By the same reasoning, increasing the excitatory weight strength should therefore increase the chance of correct activation of destination cell assemblies.

A range of connection strengths from 0.01 to 0.4 were tried for networks of 20 primitives (in which nine 2-3 cell assemblies can be constructed). As in most experiments described in this thesis, each primitive contained 150 cells. It was found that parameter values used in previous experiments, shown in table 3.1 on page 67, (with the exception of the excitatory weight strength between cells in related primitives) gave satisfactory performance provided that the number of connections was scaled up in line with the number of primitives. Excitatory weight strengths between cells in unrelated primitives were set to 0, and inhibitory weight strengths to -1. Although better performance could probably have achieved by rerunning the genetic algorithm, re-using parameters saves time and demonstrates that compound cell assemblies can exist in a network even with sub-optimal parameter values. One would expect that suitable parameter values would be similar to those for previous experiments as they also involved 2-3 compound cell assemblies. In all cases, connections between cells were assigned randomly, and not "targeted" at cells in particular primitives.

Each 2-3 cell assembly was tested in 10,000 runs, in which two of the three component primitives were activated at random. Each run had three possible outcomes:

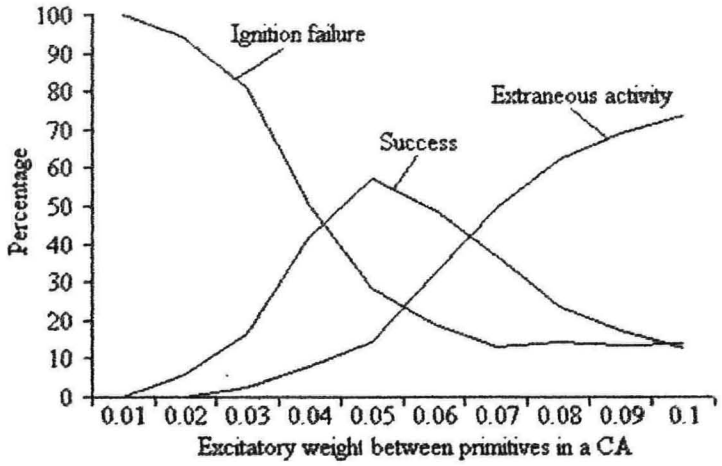


Figure 5.10: The behaviour of a network of 20 primitives over a variety of excitatory connection strengths. Each cell has 133 connections, equivalent to 20 connections per cell for each 3 primitives. At low connection strengths, the two activated primitives almost always fail to ignite the third. At high connection strengths, primitives other than those in the 2-3 cell assembly almost always ignite. The greatest chance of success, admittedly low, lies with medium connection strengths (about 0.05).

success (in which the two activated primitives ignite only the third), ignition failure (in which the two activated primitives failed to activate the third, or their own activity level died away) or extraneous activity (in which primitives other than those in the 2-3 cell assembly were activated). As in all experiments, a primitive was said to be active if ten (a nominal figure) of its cells fired. Figure 5.10 shows the result when each cell had 20 connections per 3 primitives in the network. This gave a scaling factor of 6.67 connections per cell for each primitive in the network.

Increasing the number of connections per cell produces a similar pattern to that shown in Figure 5.10. Figure 5.11 shows that the prediction concerning the relationship between excitatory weight strength, number of connections and probability of success was largely borne out. As the number of connections increases, the peak in the success curve occurs at higher weight values. Another point of interest in the graph occurs when 40 connections per cell per primitive are used and the excitatory weight passes the value of 0.2. At this point, the average energy passed to each cell becomes positive, as explained in section 4.4 on page 88, and this alters the dynam-

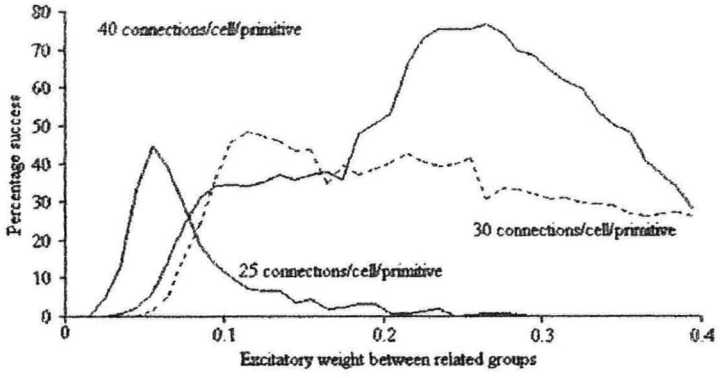


Figure 5.11: Increasing the number of connections between cells causes the chance of successful ignition to occur at higher weight values.

ics of the activation of the network. The percentage success rate suddenly rises as successful ignition no longer relies on lucky chance but on the correct energy transfer between the primitives, a more reliable mechanism.

The behaviour of the system when each cell has an average of 30 connections to each primitive is interesting. Weight connections of about 0.2 produce noticeable fluctuations in the success rate. Investigation shows that this marks a point in the parameter space where the system moves between activity generally fading out and activity generally being sustained. At this point, the system is particularly sensitive to fatigue. Increasing the weight slightly can cause relatively large numbers of cells in a target primitive to fire at the same time step, and hence to fatigue at the next time step. Large swings in the number of cells that fire at any time step increase the chance that a target primitive cannot maintain a sufficient number of cells firing to remain active.

It is possible to achieve even higher success rates by making the inhibitory connections between cells in unrelated primitives more extreme. Replacing −1 with −2 or even −4 virtually guarantees that no primitive ignites spuriously. In this case, it was found that setting inhibitory connections between cells in unrelated primitives to −2 was sufficient to squash any unwanted activity in unrelated primitives. The results are shown in Figure 5.12.

The success rate does still fall considerably as the excitatory weight rises beyond

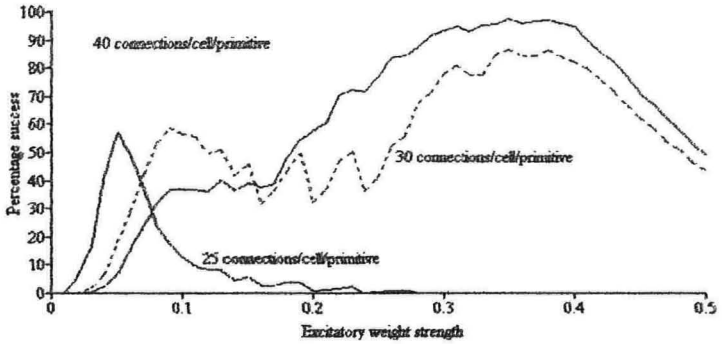


Figure 5.12: Higher success rates occur when there is mutual inhibition of primitives in unrelated cell assemblies.

the peak in the graph, but investigation shows that this is now due to the fact that at higher weight values it is more likely that one primitive will provide sufficient energy to activate a 2-3 cell assembly. Strong inhibitory weights effectively removes one possible source of error, and yet comparing figures 5.11 and 5.12 shows that they are very similar. This indicates that erroneous ignition of a 2-3 assembly by one primitive is the major cause of failure in this experiment.

The results in figures 5.11 and 5.12 appear to contradict those in chapter 4 on page 72 in which best success rates were achieved with excitatory weight strengths centred closely around approximately 0.08. It would appear that in larger networks, increasing connection strength and number of connections per cell does result in more reliable ignition. I can offer no satisfactory explanation for this apparent inconsistency at the moment.

Experiment 5.2.2(b) investigated the number of 2-3 cell assemblies that could be stored and activated reliably in a grid containing a given $N$ primitives, each of 150 cells. Parameter values were chosen to maximise the likelihood of success, *i.e.* each cell was given $40N$ connections to other cells, excitatory weights between cells in primitives within 2-3 cell assemblies were set to 0.35 and inhibitory weights between unrelated 2-3 cell assemblies were set to –2. With this arrangement, the chance of the network behaving correctly for a given pattern of activation was 95% or greater.

$N$ started at 10 and was incremented in steps of 10. For each value of $N$, 10,000 trials were carried out in which either a single primitive or two primitives were ac-

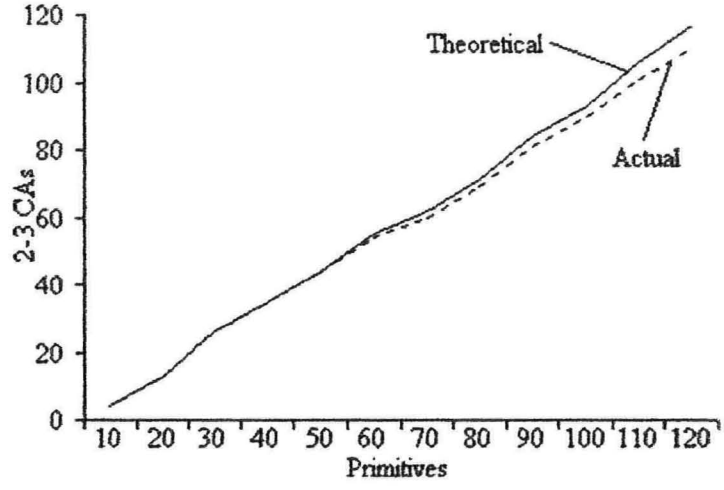


Figure 5.13: The numbers of 2-3 cell assemblies that can be stored given N primitives according to the topology illustrated in figures 5.3 to 5.5, and the actual number that can be activated on 95% or more occasions.

tivated. In the case of two primitives, 70% of the time they were present in the same 2-3 cell assembly, and the rest of the time they were unrelated. The number of trials in which the network behaved correctly was noted, *i.e.* one single primitive or two unrelated ones produced no further activation, but two related primitives produced sustained activation in a third. Figure 5.13 shows the number of 2-3 cell assemblies that can be stored for any value of $N$, given the topology described in figures 5.3 to 5.5, and the actual number that could be activated correctly and without any erroneous activation on 95% or more occasions. Both lines are approximately straight, showing that the number of primitives reliably activated is $O(N)$, but there are slight variations caused as a result of extra layers of compound assemblies being added (section 5.2.1 on page 125).

For low values of $N$ all 2-3 cell assemblies present can be activated reliably. As $N$ increases, there are a greater percentage of failures. Investigation shows that this is mainly due to increased probability of erroneous activation of a primitive which shared a compound assembly with one of the two externally activated primitives. Primitives that did not share a compound assembly with either of them were never ignited, thanks to their strong inter-primitive inhibitory connections.

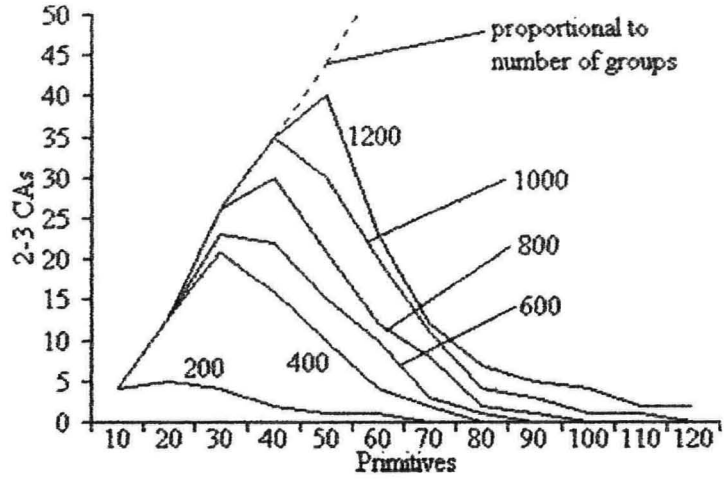


Figure 5.14: For constant numbers of connections per cell, the number of 2-3 cell assemblies that can be activated successfully and reliably diminishes rapidly with increasing numbers of primitives in the network. The dotted line shows the corresponding number of cell assemblies when the number of connections is allowed to increase in line with the number of primitives, as shown in figure 5.13.

Experiment 5.2.2(c) repeated experiment 5.2.2(b) except that each cell was given a maximum number of connections, regardless of how many primitives were stored in the network. The number of connections from a cell increased as in experiment 5.2.2(b) until this maximum was reached, whereupon it remained constant. The average number of connections from each cell to any given primitive therefore reduces as the number of primitives increases beyond this maximum point. Clearly, this reduces the number of 2-3 cell assemblies that can be successfully activated, as shown in figure 5.14. The purpose of the experiment was to determine whether a 3-4 network with reduced numbers of connections could still be used successfully to store and retrieve patterns. Clearly, such a reduced network would require less memory storage and processing power. Furthermore, although neurons in the brain can each have several thousand synapses, they can only be connected to a tiny proportion of the approximately $10^{10}$ cells in the brain. The brain is therefore *sparsely connected*, and has more in common with the network in experiment 5.2.2(c) than it does with that in experiment 5.2.2(b).

Each constant number of connections proves sufficient to maintain perfect success up to a certain number of primitives. This is justified by the fact that each constant number matches the number of connections for the same number of primitives in experiment 5.2.2(b) up to a given number of primitives. After this limit, the connections from each cell are spread more and more thinly, and it becomes ever harder for two primitives to activate a third. There is no single number of primitives that causes sudden failure, more of a gradual decline in the success rate.

It might be argued that increasing the excitatory connection strength between cells in different primitives would be sufficient to improve the performance of the network. However, pressures of time prevented me from carrying out the experiment. The problem of the limited number of connections would just manifest itself at higher numbers of primitives. The fact remains that to maintain the same degree of connectivity between primitives as the number of primitives increases indefinitely, the number of connections must increase indefinitely also. I suspect that, given the balance between the different parameters demonstrated in previous experiments, simply increasing the connection strength between cells would lead to a lowering of the success rate. Nevertheless, this approach is one that is worthy of further investigation.

## 5.3 3-4 cell assemblies can hold $O(n^2)$ stable states

The following argument may be proposed to show that a network of cell assemblies can be in any one of $O(n^2)$ stable states, in terms of the number of distinct patterns of activation that can be programmed into the network and correctly retrieved, where $n$ is the number of cells in the network. This implies that with increasing numbers of cells, the number of stored assemblies will pass the number of cells. This idea is not new, and has been alluded to by Wickelgren [186].

Again, the network is divided into a series of primitives, each of $n$ cells, in which primitive cell assemblies are entirely self-contained. They are capable of sustaining activity independently of any other primitive cell assembly and no cell takes part in more than one primitive. Furthermore, these primitives are classified into four non-overlapping primitives, termed $A$, $B$, $C$ and $D$, each containing $k$ primitive cell

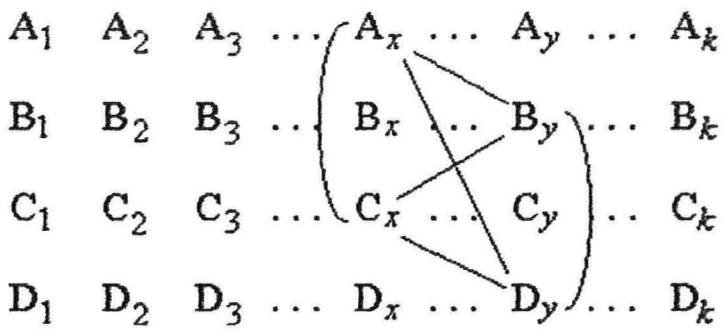


Figure 5.15: Arrangement of primitive cell assemblies in a general pattern representing $A_xB_yC_xD_y$. Solid lines represent strong excitatory connections. Although the figure shows $y$ larger than $x$, this does not have to be the case.

assemblies, a subscript differentiating the members of each primitive ($A_1$, $A_2$, etc). This arrangement is shown in figure 5.15.

3-4 cell assemblies are developed with the general relationship $A_xB_yC_xD_y$, e.g. $A_2B_7C_2D_7$, where $y$ can be equal to $x$. Connection strengths are such that activating any three of the primitives is sufficient to ignite the fourth, but activating fewer than three is insufficient to ignite any of the others. Since both $x$ and $y$ range from 1 to $k$, the network gives a total of $k^2$ compound cell assemblies. However, in order to avoid conflicts between compound cell assemblies it is necessary to have strong inhibitory connections between all primitives designated by the same letter (*e.g.* between all $A$s). Furthermore, all $A$ primitives must have strong inhibitory connections to all $C$ primitives with a different index number, and similarly for the $B$s and $D$s. This means that each primitive must have connections to each other primitive, and the number of connections increases greatly as further primitives are added to the network. The space requirement is therefore roughly proportional to the total number of connections in the network, in turn proportional to the square of the number of cells in the network. If a combination of primitives that did not correspond to a stored 3-4 cell assembly was activated externally, the inhibitory connections would ensure that some or all of the primitives were deactivated, and that there would be no extraneous firing of compound cell assemblies. An implication of this is that no more than one compound cell assembly can be active at any one time.

Activating sufficient primitives to ignite a compound cell assembly automatically inhibits all the others.

## 5.4 Proof of $O(n^2)$ stable states

In this section, I show that a network of $K$ primitives can hold on the order of $K^2$ stable states. In principle $K$ primitives can be arranged to hold $K^3$, $K^4$ or even higher powers of stable states, but these do not all represent states in which useful information can be stored. Indeed, it is shown in section 5.7 on page 151 that there are severe restrictions on the number of patterns that can be used to store information. If there are $k$ primitives of each type ($K = 4k$), there are $k^2$ ways of formulating 3-4 cell assemblies corresponding to the pattern $A_xB_yC_xD_y$. There are three requirements for this to be possible:

1. Activating any three primitives in a 3-4 cell assembly should ignite the fourth.
2. Activating one or two primitives in a 3-4 cell assembly should be insufficient to ignite either of the other two, although they should maintain activity.
3. Activating any three primitives that do not form part of a 3-4 cell assembly should not ignite any other primitives.

The first two requirements are fulfilled primarily by setting connection strengths within and between primitives correctly. Mutual connections between primitives of the same type (e.g. $A_x$ and $A_y$) should be strongly inhibitory so that activity in triplets of primitives such as $A_xA_yB_x$ tends to die out. A similar effect can be achieved in combinations such as $A_xB_yC_z$. Since $A_x$ and $C_z$ will never be present in the same 3-4 cell assembly providing that $x \neq z$, strong mutually inhibitory links between them prevent erroneous ignition. Strong mutually inhibitory links between $B_x$ and $D_z$ ($x \neq z$) prevent erroneous ignition when $B_xC_yD_z$ is activated. These inhibitory connections gives rise to the pattern shown in figure 5.15.

Since $K = 4k$ and $K \propto n$, storage capacity $O(k^2)$ effectively means capacity $O(n^2)$, *i.e.* it rises in line with the square of the number of cells in the network. It

should be pointed out that this configuration of primitives is not necessarily the only one that leads to $O(n^2)$ capacity, but it does allow that capacity to be demonstrated.

## 5.5 Implementing a network of 3-4 cell assemblies

This section deals with the problems encountered in creating a large network of cells to implement the structure described in section 5.4 on page 138. The problem of estimating parameter values in order to maximise the success rate as regards the correct completion of 3-4 cell assemblies and the non-ignition of primitives that do not form part of activated cell assemblies.

Parameters were initially determined for a network of four primitives, each fully interconnected with the others. This is equivalent to a network $A_nB_nC_nD_n$ with $n = 1$. Some parameter adjustment is inevitable as the size of the network increases. Clearly, the number of connections emanating from each cell must be scaled up in order to maintain the same overall connection strength between individual primitives. Ideally, the number of connections per cell would be determined individually for any given number of primitives in a network. However, due to time constraints, it is impractical to determine the parameters from scratch on a large network of primitives. For this reason, the parameters for the four-primitive network were used, and the number of connections scaled up linearly according to the number of primitives.

A genetic algorithm was used to determine the nine parameters listed in table 5.2 on page 140, with precision limited to three decimal places. Throughout the evolution process, emphasis was on suppressing any unwanted activity in primitives. The genetic fitness of each of the 100 population members was based on two types of behaviour. Firstly, 10,000 trials were carried out in which two of the primitives were activated externally. Any combination of parameters that produced activity in more than 10 cells simultaneously in either of the two primitives that should not ignite was rejected immediately. Any population member that survived this trial underwent a second one. 10,000 trials were carried out in which three randomly chosen primitives were activated within the same compound CA. The fitness measure was the sum of the activity in all the nodes throughout this second set of trials. In this way,

| Parameter | Value |
|---|---|
| Number of connections per cell per primitive | 24 |
| Fatigue rate | 0.902 |
| Recovery rate | 0.924 |
| Firing threshold | 0.188 |
| Retention rate | 0.798 |
| Weights from excitatory cells within a primitive cell assembly | 0.171 |
| Weights from inhibitory cells within a primitive cell assembly | −0.01 |
| Weights from excitatory cells between primitive cell assemblies | 0.077 |
| Weights from inhibitory cells between primitive cell assemblies | −0.01 |

Table 5.2: Parameter values for the 3-4 network containing four primitives.

the evolution process favoured ignition of the fourth primitive and persistence in all primitives after external activation was removed, while guaranteeing no erroneous ignition if fewer than three primitives were activated.

The genetic algorithm was repeated twenty times from different random positions and the parameter set that gave the best performance was chosen. It was not possible to ensure that this was truly the best parameter set available since each trial of the program contained a high stochastic element. However, taking the mean success rate for 10,000 runs each of 300 time steps showed that implementing these parameters in a network caused three primitives to ignite the fourth in 99.6% cases, and no completion of the 3-4 cell assembly when two primitives are activated. These parameters were then applied to larger networks, the only change necessary being to increase the total number of connections emanating from each node to maintain the correct ratio of connections to primitives. Correlations between these parameters were discussed briefly in section 3.4 on page 65.

An experiment was carried out for networks with increasing number of primitives in the arrangement specified in Figure 5.15. Possible 3-4 cell assemblies were chosen and in half the trials three of the constituent primitives were activated externally. In the other half, only two of the primitives were activated. In the former cases, success was achieved if the fourth primitive ignited. In the latter cases, success was

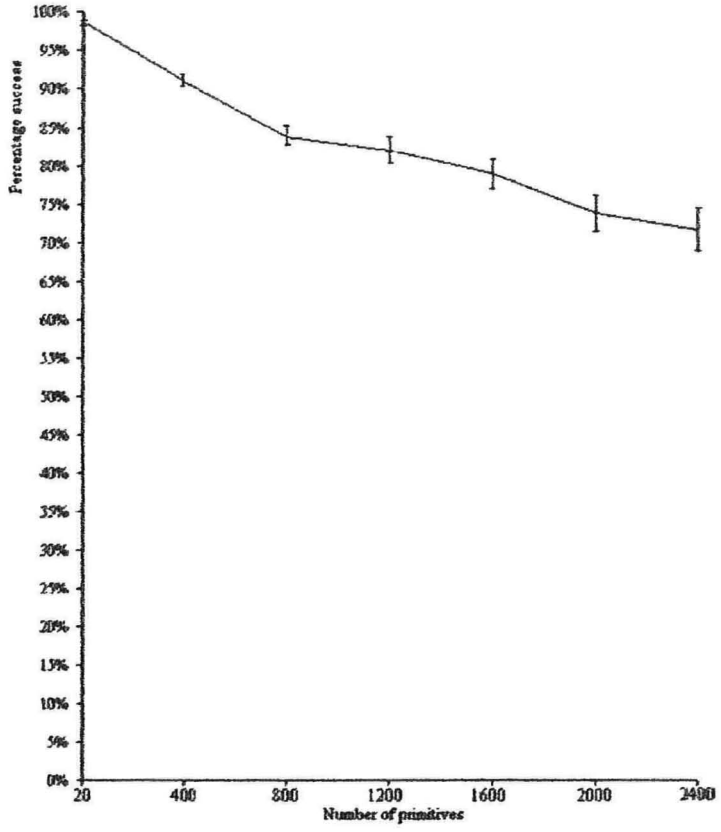


Figure 5.16: Combined success rate for large networks of primitives. The mean success rates for 10,000 trials for each number of primitives are given with error bars showing one standard deviation from the mean rate in each case.

achieved if neither of the other two primitives ignited. While it was possible for smaller numbers of primitives to try every single compound cell assembly, it became impractical to do so with larger number of primitives ($k >\sim 70$), so 50% of the cell assemblies were chosen at random and tested. Figure 5.16 shows the success rate for each number of primitives.

Figure 5.16 shows a high success rate for relatively low numbers of primitives, but a steep and steady decline as the number increases. Analysis showed that the majority of the failures occurred due to the fourth primitive of a cell assembly not igniting when presented with the other three. Relatively few of the errors occurred as a result of erroneous ignition of primitives unrelated to a 3-4 cell assembly being ignited as the strong inhibitory weights between unrelated primitives were high and such

errors disappeared completely when the weights were made even more extreme. For instance, activating primitives $A_1$, $B_2$ and $C_1$ is unlikely to cause erroneous ignition of the unrelated primitive $A_{10}$. Although there are excitatory connections from $B_2$ to $A_{10}$, their effect is clearly swamped by the inhibitory connections from $A_1$ and $C_1$. However, these extreme inhibitory connections have no effect on the intended fourth primitive, $D_2$, as none of the three activated primitives has inhibitory connections to $D_2$. The inhibitory connections between unrelated primitives can therefore be made extreme with no lowering of the success rate.

Since the success rate was higher for networks containing only four primitives, it is clear that the distribution of connections amongst the larger number of primitives must be responsible for the loss in success. The average number connections from any primitive to any other primitive stays the same as the size of the network increases. However, since connections are assigned randomly, even distributions become less likely with increasing network size.

Visual inspection of the distributions of numbers of connections to each cell show that they follow an approximately normal distribution. Figure 5.17 shows the standard deviation of the number of connections between any two primitives for a given network size. Larger standard deviations increase the probability of failure: too many connections between primitives results in ignition when two or fewer contributing primitives are activated; too few results in a failure to ignite even for three correct contributing primitives. I believe that the increasing standard deviation with increasing network size is responsible for the falling success rate.

There is no reason why cell assembly simulations should not be extended to higher order patterns. For example, primitives can be arranged to store 5-6 patterns, *i.e.* compound cell assemblies consist of 6 primitives, any 5 of which are necessary and sufficient to complete the cell assembly. The connections for such a network would match those shown for the Hopfield net in figure 5.23. However, the amount of effort and resources required to estimate the parameters and to conduct exhaustive tests renders an experiment to test such a network impractical in the limited time available.

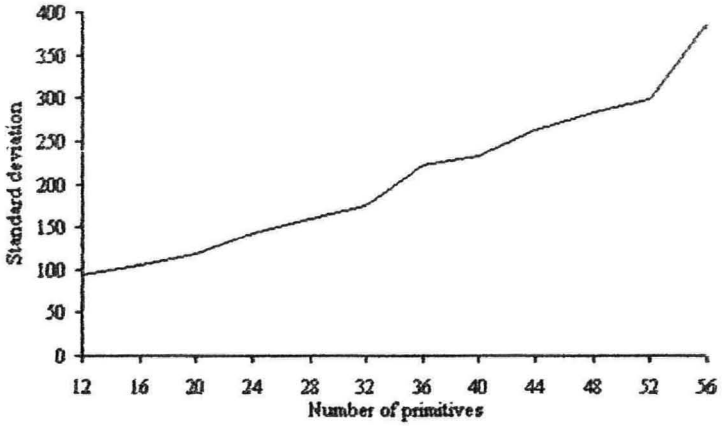


Figure 5.17: The standard deviation of the number of connections between two primitives increases as the network size increases.

## 5.6 Implementing an equivalent network using Hopfield nets

For completeness, a network similar to the one described in section 5.5 on page 139 was implemented using Hopfield nets in place of cell assemblies. I refer to the networks created as cell assemblies since Hopfield nets are sufficiently similar to the small cell assembly networks described in chapter 4 on page 72 and the term cell assembly is sufficiently flexible.

Weight estimation is a straightforward process and does not require the standard algorithm shown in table 2.1 on page 40 providing that the binary states of the cells are 0 and 1, and a threshold is introduced for each cell such that the output of the cell is 0 unless the total weighted input to the cell exceeds the threshold. Indeed, experiments show that the standard Hopfield net (binary states {1,-1}, threshold of 0 and trained using the standard algorithm) appears to be incapable of completing the 3-4 cell assembly task, as shown in figure 5.18. Figure 5.18(a) shows the equivalent of two active primitives, a situation that is supposed to persist while not allowing completion. Figure 5.18(b) shows three active primitives, a situation that has the same Hamming distance between that in 5.18(a) and four active primitives (the complete cell assembly). The standard Hopfield net must be trained not to accept three active

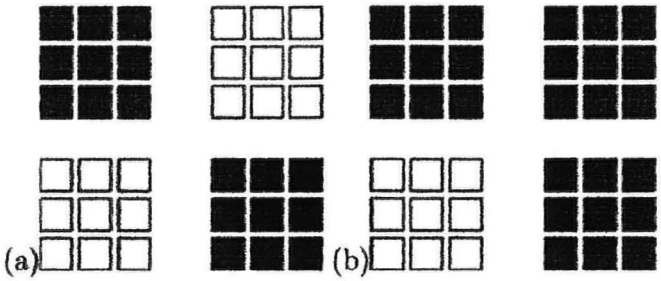


Figure 5.18: Activating two primitives should lead to a stable but incomplete pattern (a). However, activating three primitives (b) provides the network with no incentive to iterate to completion rather than towards the pattern in (a).

primitives as a stable patterns, and yet there is no reason to activate the fourth, since the two-complete state is equally near. In practice, the pattern in 5.18(b) is treated as a spurious training pattern and does not alter from one iteration to the next.

Sections 5.6.1 onwards show how a Hopfield network can be set up that does not use the standard weight setting procedure shown in table 2.1, which can still store an arbitrary number of stable patterns. Furthermore, it is not necessary with this network to account for all incomplete patterns that may legitimately occur, such as those shown in figure 5.18(a). Such patterns presented to the network simply persist, as for the cell assembly network. The network follows the topology of the 3-4 cell assembly network, with groups of cells corresponding to the primitives arranged in a number of rows. For convenience, I have retained the same terminology. Indeed, the concept of a cell assembly simulated in computer software is so vaguely defined that there is no reason not to refer to these Hopfield nets as networks of cell assemblies.

### 5.6.1 Storing $O(n^2)$ patterns in a Hopfield net

In order to keep the running time of the program to a minimum, each primitive of cells in the Hopfield net contained three cells, as opposed to 150 in the cell assembly equivalent. This is made possible by the fact that Hopfield nets contain no stochastic element, so their behaviour even at small scales is entirely predictable. Three cells is the minimum number of cells for a Hopfield net to perform completion in the same way that a cell assembly does. Although it is possible to construct the architecture

with only two cells per primitive, activating either one of them does not cause the second one to become active.

As in figure 5.15, primitives were arranged in rows of four corresponding to $A$, $B$, $C$ and $D$. Weights between cells within the same primitive can be set sufficiently high so that activating a sufficient number of cells within the primitive is sufficient to activate the others. This is achieved by implementing a threshold, similar to the firing threshold used by cell assemblies. If the total incoming activity to any cell is greater than this threshold, its state is set to 1. Otherwise, the state remains at 0, as shown by equation 5.1.

$$a_{j,t} = \begin{array}{ll} 1 & \text{if } \sum_i w_{ij} a_{i,t-1} > \theta \\ 0 & \textit{otherwise} \end{array} \tag{5.1}$$

where $a_{i,t}$ is the activity of cell $i$ at time $t$, $w_{ij}$ is the weight from cell $i$ to cell $j$. There is, of course, no weight from any cell to itself ($w_{ii} = 0$) to prevent any cell from remaining active unless its activity is reinforced by other members of the same primitive. Setting a threshold of 0.5 and each weight within the primitive to 0.6 allows the primitive to ignite if any of its cells are active. Setting weights between cells within any 3-4 pattern to 0.07 allows three primitives to activate cells in the fourth but not two primitives. Each active primitive always completes itself, so it provides a total of 0.21 units of activity. Three contributing primitives therefore provide 0.63 units, which exceeds the threshold, whereas two contributing primitives only provide 0.42 units.

An exhaustive trial shows that a Hopfield net constructed in such a manner gives flawless recall of stored 3-4 cell assemblies even when large numbers of primitives are used. In most determinations of the capacity of a Hopfield net, (chapter 2 on page 31), some degree of error in pattern recall is assumed, usually 1% of the bits of the recalled pattern. This experiment made no such assumption. Figure 5.19 shows the recall of one such pattern.

The behaviour of this system appears to negate Amit's [6] proposition that Hop-

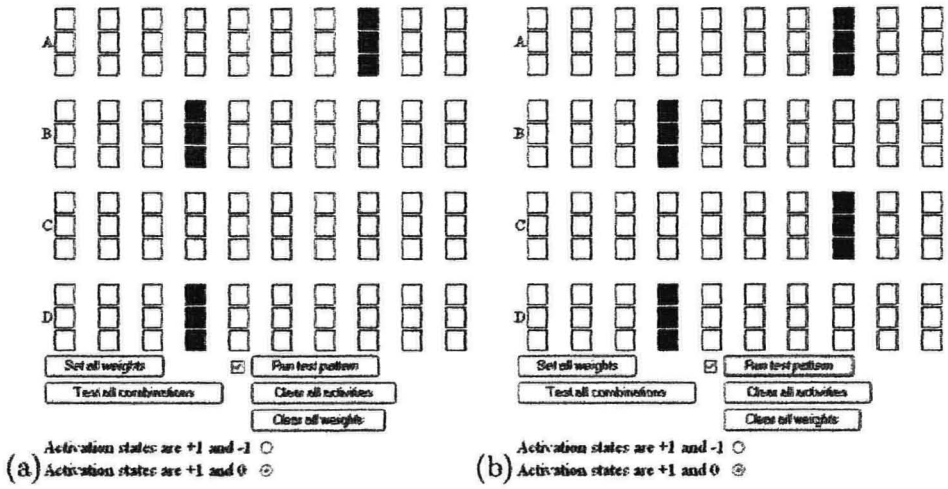


Figure 5.19: Presenting three constituent primitives in a 3-4 pattern (a) leads to pattern completion (b).

field nets can only store on the order of $N$ patterns for $N$ cells. However, Amit specifies that the patterns stored must be *stable*, *i.e.* they must represent the bottoms of attractor basins. One flaw with this system of weights is that the network is prone to runaway iguition if primitives $A_x$, $B_y$ and $C_z$ ($x \neq y \neq z$) are activated concurrently. On the first iteration, one 3-4 pattern (either $A_xB_yC_xD_y$ or $A_zB_yC_zD_y$) is erroneously activated, and on the next iteration, activity spreads to the entire network. However, this problem is corrected by setting weights between cells corresponding to the same letter but in different columns to strong mutually suppressive values (-10 works well). In this case, activating $A_x$, $B_y$ and $C_z$ causes one and only one erroneous $D$ primitive to activate without any loss in ability to recall stored patterns. Similarly such weights prevent more than one 3-4 pattern from being active at any time (figure 5.21).

Further improvements can be made. In each of the stored patterns $A_xB_yC_xD_y$, the index of the $A$ primitive matches that of the $C$ primitive, and similarly for $B$ and $D$. Any retrieved pattern in which this is not the case represents an error, which can be prevented by setting strong inhibitory connections between all $A$ cells and $C$ cells in different columns, and similarly for $B$ and $D$ cells. This does indeed remove spurious 3-4 patterns, although activating $A_xB_yC_z$ does still lead to a stable

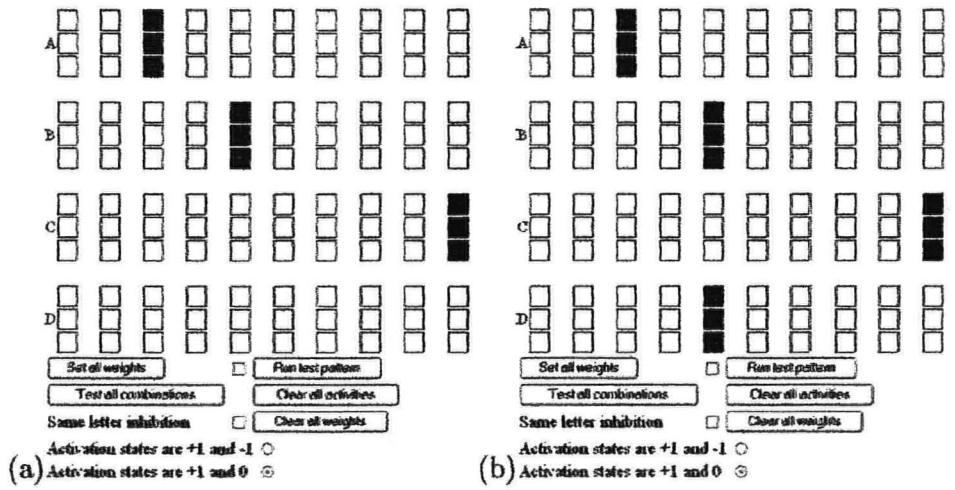


Figure 5.20: Presenting three primitives not part of a 3-4 pattern (a) leads to at most one erroneous ignition (b).

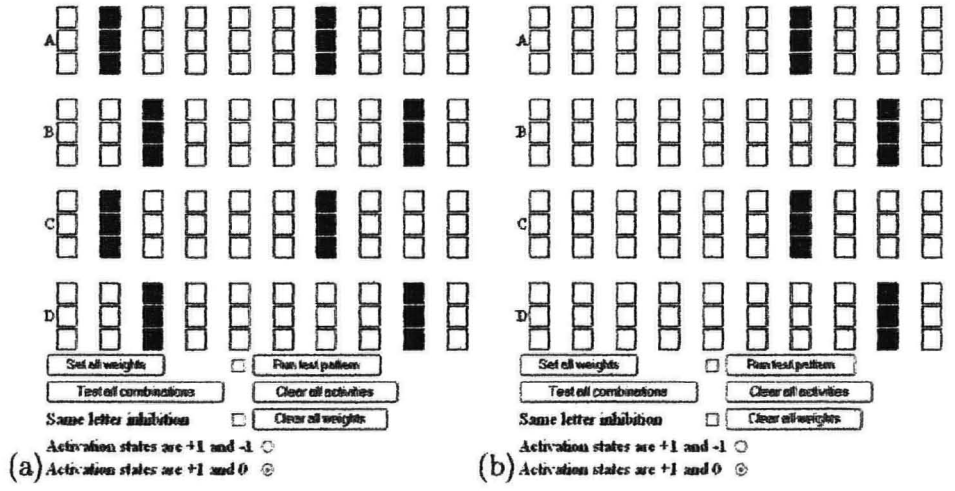


Figure 5.21: Mutually inhibitory connections between cells representing the same letter in different columns prevents more than one 3-4 pattern from being active simultaneously. Activating two such patterns (a) leads to one shutting down on the next time step (b).

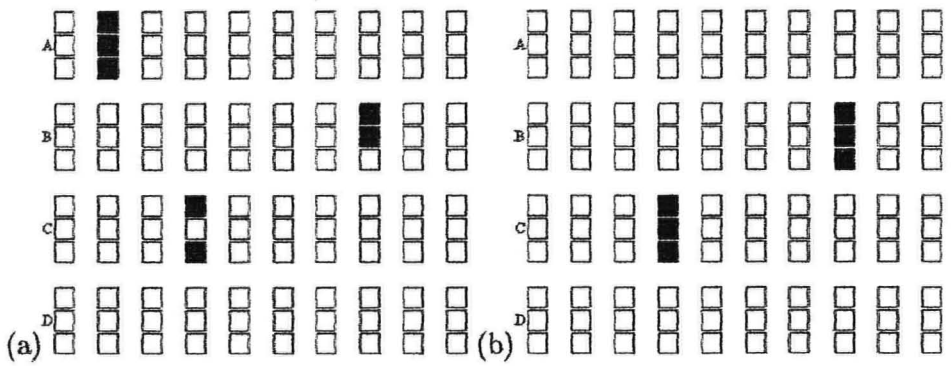


Figure 5.22: Inhibiting connections between $A$ and $C$ cells in different columns, and between $B$ and $D$ cells in different columns prevents erroneous activation of 3-4 patterns. (a) represents the test pattern, and (b) the stable state that it produces.

attractor basin not representing any stored pattern, as shown in figure 5.22. In figure 5.22(a), representing an initial activity state of the network, two of the primitives are only partially activated. Figure 5.22(b) represents the stable state reached by the network and shows that these primitives have completed. The inhibitory connections between the $A$ and $C$ cells have prevented them both being active at the same time, but, interestingly, it was the incomplete $C$ primitive that shut down the complete $A$ one. This is a side-effect of the fact that the cells in the network are updated in a fixed order, so that activity to the right of the grid tends to shut down activity to the left, and activity lower down the grid tends to shut down activity higher up.

### 5.6.2 Storing $O(n^3)$ patterns in a Hopfield net

The network can be extended to give a performance that appears at first sight to store on the order of $n^3$ patterns. Primitives are arranged in six primitives ($A$ to $F$). Compound "cell assemblies" are set up according to the pattern $A_x B_y C_z D_x E_y F_z$ as shown in figure 5.23. Setting weights between cells in a compound cell assembly to 0.034 is sufficient to allow five primitives to activate the sixth, but not four (figure 5.25). The fact that there are $k$ values of $x$, $y$ and $z$ suggests that a network of $K$ primitives can store of the order of $K^3$ compound patterns, and hence $O(n^3)$, where $n$ is the number of cells in the network. Exhaustive experiments do show that this

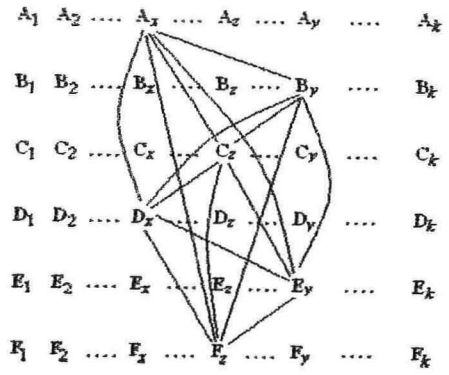


Figure 5.23: A Hopfield net configuration for 5-6 patterns.

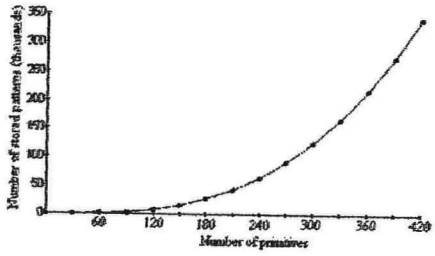


Figure 5.24: Hopfield nets can be configured to have a capacity of $O(N^3)$.

is the case, as illustrated by figure 5.24. The number of patterns that can be stored is $(\frac{K}{6})^3$ as the primitives must be arranged in groups of 6 for each compound cell assembly.

These results are significant because they appear to contradict researchers such as [6] who state that the storage capacity of a Hopfield is linearly proportional to the number of cells. However, the patterns shown so far represent *stable states* rather than states in which useful information can be stored. There is no reason why the architecture cannot be extended to provide the equivalent of 7-8 cell assemblies, with storage of the order of $n^4$ patterns etc. In each case, the weights between cells within the same pattern are $\theta/(\nu.q) + m$ where $\theta$ is the cell threshold, $\nu$ is the number of cells in each primitive, $q$ the number of active cells required to complete the pattern (*i.e.* 7 in the case of 7-8 patterns) and $m$ is some arbitrarily small value (e.g. 0.001) sufficient to push the total weighted input to each cell just above the threshold. As the order of the network ($q$) increases, each of the weights decreases, and so does the

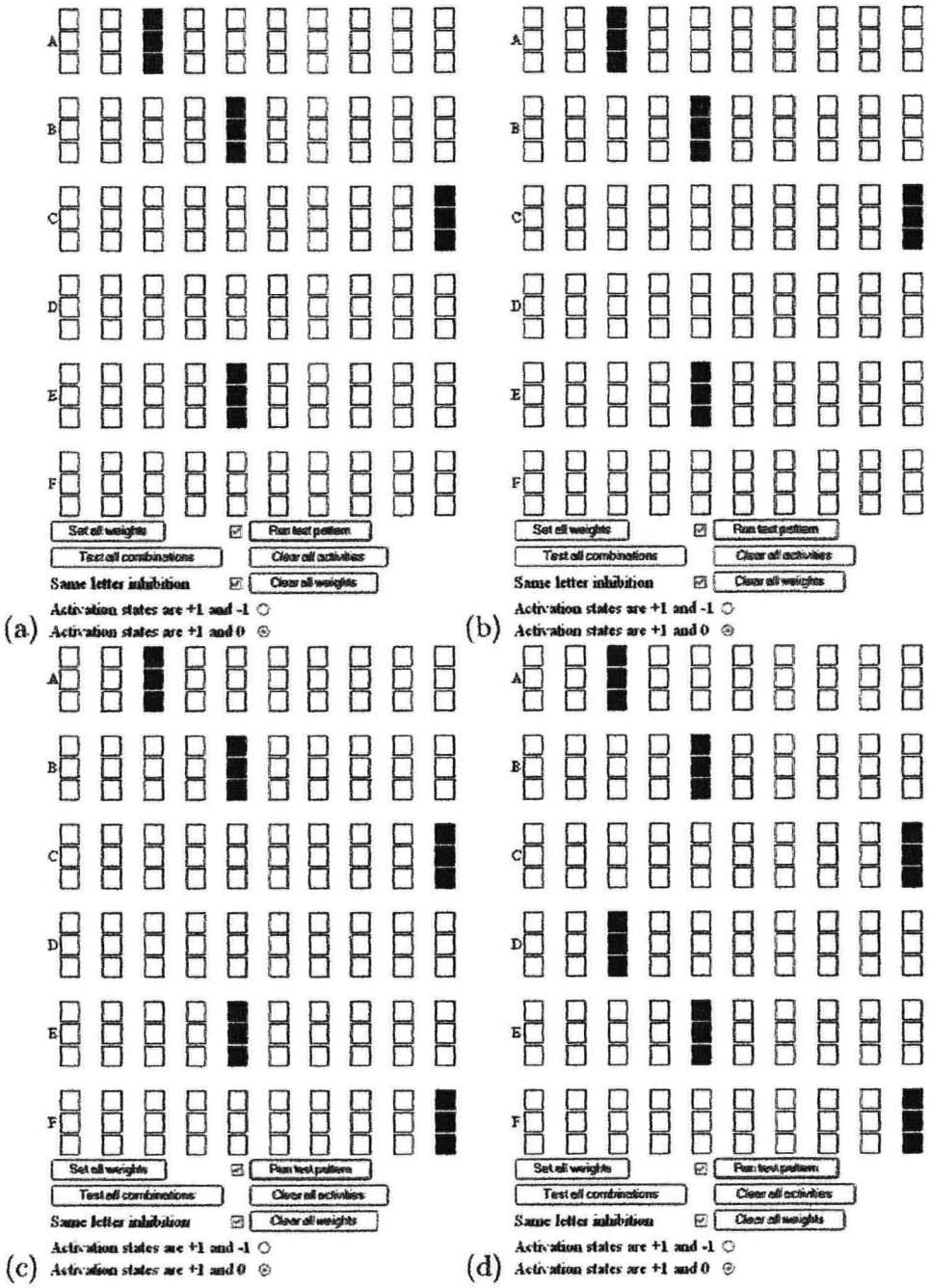


Figure 5.25: A Hopfield net can be created to store and complete 5-6 patterns, such that activating four primitives (a) is insufficient to ignite any others (b), but activating five (c) is sufficient to complete the pattern (d).

margin of error between the summed input from $q-1$ cells and $q$ cells. For a Hopfield net with preset weights and one connection only from any cell to any other, this does not provide a problem. For a cell assembly, with a heavy stochastic element, the margin of error is quickly swamped by fluctuations in the weighted input. For this reason, while higher-order Hopfield nets are quite feasible, the same cannot be said with certainty for higher-order networks of cell assemblies. This is illustrated by the fact that for the Hopfield net, when $k = 12$, the number of stored 3-4 patterns in the Hopfield net equals the total number of cells in the net (144), and at this point, the net has a perfect recall of all the patterns. With 150 cells in each primitive, the 3-4 cell assembly network does not reach the same point until $k = 600$ (corresponding to 2400 primitives all together), and in this case, the recall rate is only 71.6%, as shown in graph 5.16.

## 5.7 5-6 Hopfield nets do not break the rules of Information Theory

At first sight, it may appear that Hopfield nets, arranged to store 5-6 patterns offer an unlimited amount of storage. Information Theory [161, 162] states that a purely random sequence of $n$ bits cannot be represented by fewer than $n$ bits of information, and yet a 5-6 Hopfield net, with $n$ cells and $n(n-1)$ connections (no cell has a connection to itself) can store $n^3$ patterns. However, in this section, I show that this property disappears as soon as any attempt is made to store only certain patterns in the net and not others. This does not render the 5-6 nets useless, but it does help to keep their abilities within perspective.

The stable states of the 5-6 Hopfield net are represented by the 5-6 patterns themselves. Activating fewer than five primitives in a 5-6 pattern does not cause completion and therefore provides no information, and yet activating more than six primitives causes at least some of them to be shut down due to the strong inhibitory connections between patterns. The 5-6 Hopfield net can therefore only be used to hold information by storing some patterns and omitting others.

Consider a situation in which pattern $A_xB_yC_zD_xE_yF_z$ is to be omitted, perhaps

to represent a '0' bit in a bit sequence. The connections that make up this particular pattern ($A_x - B_y$, $A_x - C_z$ etc.) also form part of many other patterns e.g. $A_x - B_y$ is present in $A_xB_yC_wD_xE_yF_w$. If any of these patterns is to be stored (perhaps representing '1' in the bit sequence), then the connection is set. Since the connections between cells $A$, $B$ and $C$ are duplicated in cells $D$, $E$ and $F$, it only takes three patterns to be stored for the pattern $A_xB_yC_zD_xE_yF_z$ to be stored erroneously, as shown in figure 5.26, which shows how storing patterns $A_1B_2C_4D_1E_1F_4$, $A_4B_2C_3D_4E_2F_3$ and $A_1B_4C_3D_1E_4F_3$ leads to the erroneous storing of pattern $A_1B_2C_3D_1E_2F_3$. Every excitatory connection shown in figure 5.26(d) is duplicated somewhere in figures 5.26(a) to (c).

It is still possible to construct a set of network connections that will store information. For instance, the connections shown in table 5.3 store the binary pattern 110001110 in a 3-4 Hopfield net. However, this message was contrived so that no erroneous information was stored. Each '1' bit corresponds to a compound cell assembly, the identity of which depends on the order in which the patterns are "read" from the network. The $C$ and $D$ indices for each cell assembly duplicate the $A$ and $B$ and can be ignored here. If the patterns are read in the order $A_1B_1$, $A_2B_1$, $A_3B_1$, $A_1B_2$, $A_2B_2$, $A_3B_2$, $A_1B_3$, $A_2B_3$, $A_3B_3$, then the bit string 110001110 is encoded by the cell assemblies $A_1B_1C_1D_1$, $A_2B_1C_2D_1$, $A_1B_3C_1D_3$, $A_3B_2C_3D_2$, and $A_2B_3C_2D_3$, as shown in figure 5.27, corresponding to the connections shown in table 5.3. Only connections strengths between cells in different primitives are shown, and each connection is symmetrical. Each tick represents a weight strength of 0.057, each cross an inhibitory weight of −10, and each 0 a zero connection. In this case, the indices of the $B$ primitives are analogous to the low digit of an "address" specifying a bit position along the message, and the $A$ indices are analogous to the high digit.

Table 5.3 does not represent the minimum number of bits that must be transmitted as it contains a great deal of redundancy. Inhibitory connections are predetermined, as are connections between cells in $A$ primitives and those in $C$ primitives, and those between cells in $B$ primitives and $D$ primitives. Weights from primitive $A$ to primitive $B$ are matched by those from primitive $A$ to $D$, from primitive $B$ to $C$ and from primitive $C$ to $D$. In fact, the only 9 weight values need to be transmitted,

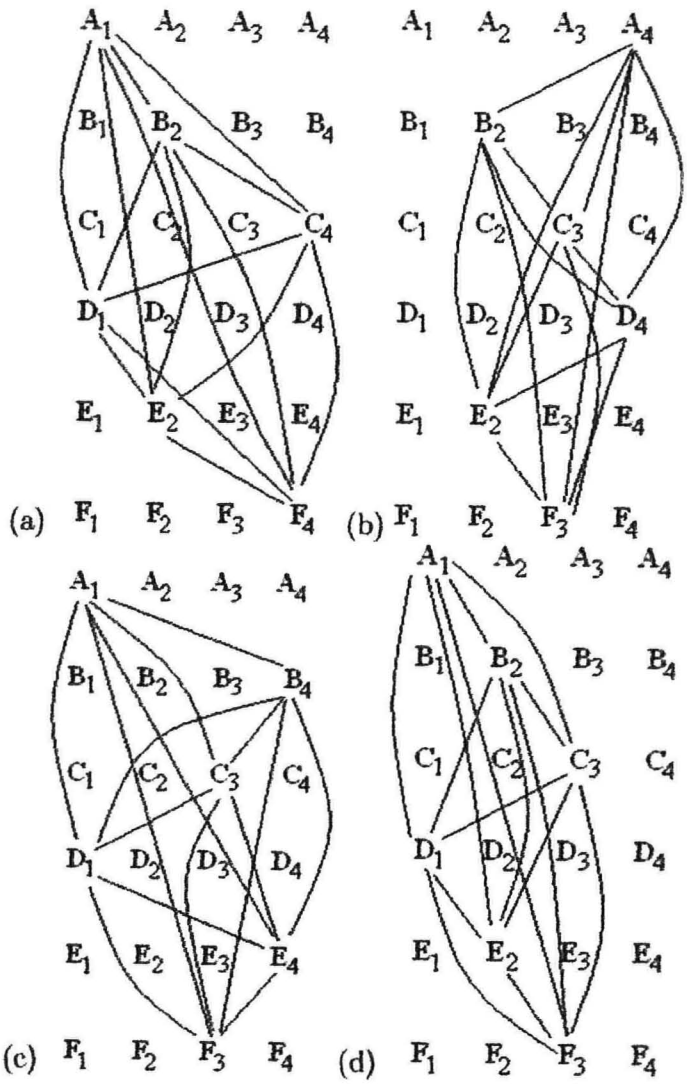


Figure 5.26: The three patterns stored in (a) to (c) lead to the erroneous storing of the pattern in (d). Inhibitory connections have been omitted from the figure for clarity.

| $A_2$ | $A_3$ | $B_1$ | $B_2$ | $B_3$ | $C_1$ | $C_2$ | $C_3$ | $D_1$ | $D_2$ | $D_3$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| × | × | √ | 0 | √ | √ | × | × | √ | 0 | √ | $A_1$ |
| | × | √ | 0 | √ | × | √ | × | √ | 0 | √ | $A_2$ |
| | | 0 | √ | 0 | × | × | √ | 0 | √ | 0 | $A_3$ |
| | | | × | × | √ | √ | 0 | √ | × | × | $B_1$ |
| | | | | × | 0 | 0 | √ | × | √ | × | $B_2$ |
| | | | | | √ | √ | 0 | × | × | √ | $B_3$ |
| | | | | | | × | × | √ | 0 | √ | $C_1$ |
| | | | | | | | × | √ | 0 | √ | $C_2$ |
| | | | | | | | | 0 | √ | 0 | $C_3$ |
| | | | | | | | | | × | × | $D_1$ |
| | | | | | | | | | | × | $D_2$ |

Table 5.3: Connections for storing 110001110

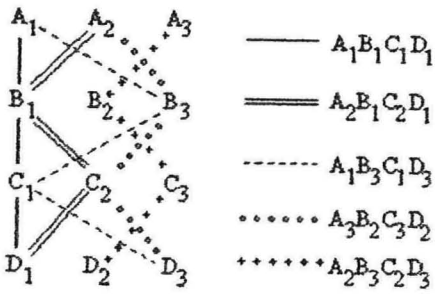


Figure 5.27: Pattern of cell assemblies to store 110001110. Not all connections are shown.

| $B_1$ | $B_2$ | $B_3$ | |
|---|---|---|---|
| √ | 0 | √ | $A_1$ |
| √ | 0 | √ | $A_2$ |
| 0 | √ | 0 | $A_3$ |

Table 5.4: Minimum connections needed to store 110001110

as shown in table 5.4, which can be encoded in only 9 bits. In fact, the encoding is a straight-forward matter, as reading down the columns of the table and interpreting a tick as 1, reconstructs the original bit string.

## 5.8 Summary

In this chapter I have proposed that in theory a network of $n$ cells can be constructed consisting of a series of cell assemblies termed *primitives*, and that this network can hold on the order of $n^2$ patterns of activity. An implementation of this indicated that small networks could be activated with almost perfect reliability, but that the reliability fell gradually as the network size increased. I also propose that in principle higher order networks of compound cell assemblies can be constructed, containing compound cell assemblies consisting of $K$ primitives in which $(K-1)$ primitives are needed to ignite the last one. However, time restrictions coupled with the large amount of computing power required make this impractical for the scope of this thesis.

Comparisons were made between the cell assembly network and an equivalent Hopfield net, and it was shown that a Hopfield net could be constructed which behaved in a similar manner to cell assembly network, and therefore also had the capacity to store on the order of $n^2$ bits of information. This largely agrees with the known literature, which states that Hopfield nets store on the order of $n$ patterns (the maximum number being $0.138n$), each of which is $n$ bits long. The capacity of Hopfield nets is therefore generally considered to be on the order of $n^2$ bits also, although some researchers (chapter 2 on page 31) claim that this can be extended. Furthermore, the general success of the experiments to store $O(n^2)$ patterns in cell

assemblies and Hopfield nets suggests that higher order networks can be built, that can store on the order of $n^3$, $n^4$ or more, and a Hopfield net that can store on the order of $n^3$ was constructed and successfully tested. In the last section I show that this architecture has severe limitations as an information encoder and does not live up to the promise of almost infinite information storage. Such a restriction is inevitable given the restrictions imposed by both Information Theory and, indeed, common sense. The claim that Hopfield nets can be arranged to have a capacity of $O(n^3)$ and higher, while strictly speaking true, appears to be largely an empty promise. Although the Hopfield net can indeed be arranged to have $n^x$ stable states, for any arbitrary positive integer $x$, these do not correspond to $n^x$ states in which useful information can be stored. However, I show that such a net can still be used to encode information, and investigations of the practical carrying capacity of a 5-6 Hopfield net are continuing.

The restrictions applying to Hopfield nets apply equally well to networks of cell assemblies. Although experiment 5.5 on page 139 shows that a network of 5-6 compound assemblies can in principle hold $O(n^3)$ stable patterns, these cannot all be used to store information. A practical network would require that some 5-6 patterns be present and some be absent, and would suffer from the same problem that a Hopfield net would. The problem is also exacerbated by the fact that given the implementation used, the ability to ignite the 5-6 assemblies reliably decreases approximately linearly with increasing numbers of primitives. This effect is almost certainly due to the distribution of the numbers of connections between individual primitives, which gradually increases as more primitives are added.

The question may be asked "If cell assemblies and Hopfield nets are so similar, why should we carry out research into cell assemblies?" This is a valid question. After all, the behaviour of Hopfield nets is well understood, and the experiments described in this chapter show that they can clearly reproduce much of the behaviour of cell assemblies. One valid argument returns to the point raised in chapter 1 on page 2: Cell assemblies are no less than an attempt to model structures believed to be present in the brain, whereas Hopfield nets are only loosely inspired by the brain. Of course, this point is irrelevant to any researcher merely interested in the

information processing abilities of connectionist nets, but the brain is capable of extremely complicated processing, so there may be great benefit in emulating the brain. A more relevant argument is the flexibility of cell assemblies: Activity in cell assemblies builds up, is sustained for some time, and then dies out. The cell assembly may ignite other cell assemblies, and be re-ignited itself at some other time. Experiment 4.4 on page 88, for example, shows a complex, if unintended, chain of ignition and suppression occuring in a network of only six primitives. The Hopfield net iterates until it reaches a stable state, and then remains in that state permanently. The pseudo-stable states through which a network of cell assemblies passes allow for more complex processing than the stable states of the Hopfield net.

# Chapter 6

# Conclusions and Further Research

This thesis describes work carried out to test the capabilities of cell assemblies simulated in computer memory. The cells simulated maintain some properties found in biological neurons, although the equations on which they are based have been severely simplified in order to reduce simulation running time. Even with a simple neuron model, cell assemblies demonstrate some interesting and useful properties. Later work showed that large networks of cell assemblies could be constructed, whose capacities could be predicted and verified. The networks of cell assemblies worked well for relatively small numbers of primitives, but the rate at which compound assemblies could be successfully recalled decreased in an almost linear manner as the number of primitives increased.

Cell assemblies can be formed in small networks of cells with initially random connections between the cells. The behaviour of the networks was determined by a handful of parameters, all of which interact in complicated ways. The problem of parameter selection readily lends itself to a genetic algorithm, and a variety of satisfactory parameter values was found through the application of a simple genetic algorithm. This approach proved so powerful, that it was possible to get a variety of behaviours from the cell assembly, including mimicking the behaviour of the TRACE cell assembly model [98]. Experiments were also carried out in which some parameters, namely weight strengths, were derived from a process of Hebbian learning.

Throughout the thesis cell assemblies are compared to their nearest relative in the taxonomy of neural net architectures, namely Hopfield nets. Indeed, the definition

of simulated cell assemblies is so vague that Hopfield nets may be considered to be valid implementations of cell assemblies. They certainly do demonstrate a range of properties postulated for biological assemblies, such as completion, in which some activated cells in the Hopfield net cause others to activate, and persistence, in which a pattern, once activated, persists for a long time. However, these properties are not dynamic, *i.e.* once the Hopfield net has moved to a stable state, it remains in that state indefinitely and is incapable of moving away from it until external conditions change.

This is the area in which cell assemblies are potentially superior to Hopfield nets. They are more flexible, in that their patterns are dynamic: Although the activity in individual cells may wax and wane, the overall pattern in the network (in terms of whether the cell assembly is activated as a whole) remains constant. The dynamic nature of cell assemblies allows them to ignite, and to deactivate, suddenly or over time. They can ignite other assemblies before fading out themselves, resulting in sequential processing of "ideas" in a network. Simulated cell assemblies have been used to implement concepts associated with intelligent behaviour such as counting and variable binding. In short, cell assemblies promise to be capable of a greater range of behaviour than Hopfield nets, and this is what makes them suitable for further research.

## 6.1 Conclusions

Small hierarchies of cell assemblies can be formed by a similar process of learning, such that individual primitive cell assemblies can be activated, and that compound 2-3 cell assemblies can be ignited. Each primitive consisted of 150 cells, with 10 cells being chosen as the activity level necessary for a primitive to be considered active. The parameters controlling the global behaviour of the network interact in a manner that was too complex for me to predict. It is a fairly easy matter to set up a single assembly in a network, as excitatory weights between cells can be set fairly high. The effect of fatigue does put a limit on weights, as unrestricted firing of cells leads to a much lower number firing on the next time step. However, determining

parameters for compound cell assemblies is substantially harder: Not only is there an extra parameter to calculate, the excitatory weight strength between cells in different primitives (or two extra parameters if the inhibitory weight strength need not be its negative equivalent), but the parameters are now subject to tight controls. They must be set so that $k$ active primitives are insufficient to ignite any others, but that $k+1$ primitives are sufficient to do so. As $k$ becomes ever larger, the problem becomes harder, and that has curtailed my research. For instance, for 2-3 compound assemblies ($k = 1$), parameter sets were determined both by genetic algorithm and by learning. These two turned out to be different, from which conclusions could be drawn. For 3-4 compound assemblies ($k = 2$), parameter sets were determined by genetic algorithm alone. There is no logical reason why parameters for a 3-4 network should not be learned, apart from the fact that the desired values need to be known already since they control the ratio in which patterns are presented. For the 5-6 assemblies ($k = 4$), experiments were only carried out using a Hopfield net.

Several of the findings have been counter-intuitive, the most notable of which is that the most effective performance occurs with a parameter set in which overall activity passed between primitives is *negative*, and yet activation still happens. This led to the discovery of the "lucky neuron" effect, in which a minority of the most strongly connected cells are responsible for igniting the primitive and the low activation energy passed mainly acts as a brake to reduce runaway activation. An experiment was carried out in which this brake was removed, and there was indeed uncontrolled activity in all the primitives, which led to widespread fatigue and then a crash in activity. A good performance can be obtained by arranging the frequency of training patterns so that the weights between and within primitives more closely match a more intuitive ratio, but this does not give the highest success rate.

Section 4.8 on page 106 investigates the effects of allowing cells within a network to activate spontaneously, *i.e.* in the absence of any external stimulus. It is shown that even a small spontaneous activation probability leads to the network developing some interesting properties. These occur due to the continual presence of Hebbian learning in the network even after cell assemblies have formed. It was found that cell assemblies could enlarge as a result of recruitment at their outer edges, and that

assemblies that were rarely ignited tended to be forgotten. Fractionation does occur within large assemblies, although the cause of this and circumstances in which it happens were not investigated.

When primitives have to be combined within a compound cell assembly, there are two conflicting imperatives, one tending to increase the weights, the other tending to reduce them. For the simplest compound assembly, the 2-3 assembly, any one primitive must not provide enough activation energy to the other two to allow ignition, but any two primitives must provide sufficient energy to ignite the third. Larger weight values tend to fulfil the latter requirement while rendering the former less likely. Restrictions are even tighter for 3-4, in which no fewer than three primitives are necessary to ignite the fourth. The problem is made worse by the fact that the lucky neuron effect still applies since the destinations of the connections were assigned in the same random way in all experiments.

A set of weights that gave acceptable performance for a 3-4 network was determined using a genetic algorithm. It was impractical to test whether these weights could be learned due to the prohibitive amount of memory that would be required to store learned weight values. If the 3-4 network were to follow the same pattern as the 2-3 network, then the learned weights would not give the same high performance as the evolved weight values.

Having validated the performance of the simplest possible 3-4 network, the next obvious step is to combine these networks into larger ones that can contain more than one 3-4 pattern. It was shown that a network containing a large number of primitives arranged in a 3-4 pattern had on the order of $n^2$ stable states consisting of 4 primitives such that activating no fewer than three primitives would cause the fourth to ignite. $n$ represented the number of cells in the network. Such a network was implemented both as a set of cell assemblies and as a Hopfield net, and it was confirmed that the number of stable states did indeed increase quadratically with the number of primitives, and hence the number of cells, in the network.

One justifiable criticism of these conclusions is that they apply to one particular implementation of a cell assembly, *i.e.* only one set of equations, applied in one particular order. Perhaps, if pre-not-post LTD had been applied in conjunction with

a slightly different way of calculating fatigue, then success rates may have been higher. My only justification can be that a large number of variations of the architecture were initially tried before settling on the one described. The architecture that appeared to give the best results was the one that was finally chosen.

## 6.2 Have the objectives been fulfilled?

In general, yes. Some of the conclusions have been rather disappointing, particularly those involving the capacity of networks with a large number of primitives, but the thesis has achieved everything that I set out to achieve.

### 6.2.1 Establishment of a cell assembly in a network of cells

This objective was fulfilled in chapters 4 on page 72 and 5 on page 121, but it was the main aim of section 4.1 on page 74. Experiments 4.1.1 and 4.1.2 (pages 75 and 79) show that a network, obeying a few simple equations, can develop connections between those cells forming a cell assembly. The cell assembly displays all the properties specified by Sakurai [154]. Furthermore, it has been shown that a network can develop more than one cell assembly, and that these assemblies can interact in a meaningful way.

### 6.2.2 Investigation of properties of cell assemblies

Various properties of the simulated cell assemblies have been investigated in chapter 4 on page 72. Several of these were the aims of specific experiments, such as the effect of spontaneous activation on the enlargement or fractionation of cell assemblies. Other effects were discovered in passing, such as the way in which the distribution of connections among the cells gave rise to the lucky neuron effect, and influenced the choice of parameters by the genetic algorithm. The interactions between these properties, particularly the way in which the pernicious lucky neuron effect alters behaviour such as recruitment or forgetting, has not been studied, mainly due to pressures of time.

### 6.2.3 Establishment of correlates between simulated cell assemblies and those in the brain

Although it was not my intention to support or undermine any particular theory of cell assemblies, one side effect of the research has been to provide some support for Palm's theory [139] in which he proposes that neurons in the brain are partitioned into modules. Within these modules neurons are heavily connected, but there are relatively few connections between modules. The connections between primitives mimics these modules, in which a few connections between primitives are responsible for the majority of the inter-primitive activity.

Experiments that establish networks of communicating primitives also help to answer a criticism by Milner [134], that the natural consequences of Hebbian learning would be for connections between cell assemblies to become indistinguishable from those within cell assemblies, and that consequently the assemblies would not be able to maintain their independent existence. While the genetic algorithm can throw no light on this matter, since it assumes that inter-primitive connections differ from intra-primitive ones, the experiments involving learned weights show that Milner's criticism is not well founded. The secret to differential connections is the difference in co-firing rates, itself a result of difference in frequencies of pattern presentation.

### 6.2.4 Determination of the storage capacity of cell assembly networks

This is the area of research that has produced the most disappointing results. Experiment 5.1 on page 122 did show that a network of cells could be arranged to hold $O(n)$ primitives that could be activated independently, even when these primitives overlapped to a small extent. Increasing the overlap resulted in erroneous ignition of neighbouring primitives.

Although in theory the proposed architecture of the cell assemblies in the 3-4 pattern produced a network with storage capacity of order $n^2$ bits, where $n$ is the number of cells in the network, experiments showed that my system did not match this in practice. I say "my" system, since there is no reason to believe that this failure

represents a mistake in the theory. It could well have been due to stochastic variation in the connections (a quite probable cause given the problems that that effect has caused in previous experiments) or simply due to a poor choice in the mathematical determination of the model's behaviour. Perhaps, for example, if the fatigue and/or recovery from fatigue had not been governed by constant parameters but some more complex function, then results may have been better. For the theory of $O(n^2)$ to have been fully vindicated, the line shown in graph 5.16 on page 141 should have had an average gradient of zero, *i.e.* some slight variation permissible due to the effects of noise, but generally the same success rate independent of the number of primitives in the network. Even a cursory glance shows that this clearly is not the case. This thesis must therefore end with the problem of whether a network of assemblies can have $O(n^2)$ capacity not completely addressed.

When it comes to determining the capacity of Hopfield nets, the approach explained in chapter 5 on page 121 is just one of several, each of which comes to give a slightly different conclusion. Section 2.3.2 on page 43 outlines these approaches, and shows that they all make slightly different assumptions. Chapter 5 needs to make no assumptions about probability of error, or the shape of error distributions. It is also the only approach to be applied to cell assemblies as well as Hopfield nets.

## 6.3 Further Research

This thesis has shown that cell assemblies do have potential to be a useful tool in any connectionist toolbox. However, a great deal of work is necessary before they can become as useful as more established architectures such as multi-layer perceptrons or Hopfield nets. There is very little hard-and-fast theory concerning cell assemblies, and the work described in this thesis does not improve the position much. It does, however, demonstrate that the behaviour of the network is a complex interplay of various parameter values. Nevertheless, I feel that it is open to some sort of mathematical analysis, from which, perhaps, optimal parameter values may be calculated for any configuration of primitives without the need for time-consuming genetic algorithms.

While experiments have been carried out investigating whether connections could

be learned in small networks, time restrictions have prevented implementing learning in large networks. On a personal computer, programs involving large networks of cells (up to 2400 primitives have been tested) often require more than a week of continual computer running time. This would be increased many times over if learning were to be implemented. Experimental findings show that recall rates fall away approximately linearly as the number of primitives in a large network increases, and a similar disappointing performance is to be expected from large networks in which weights are learned.

### 6.3.1 Theoretical research

Chapter 5 demonstrates that 3-4 networks, both in the form of cell assemblies and Hopfield nets, can be used to store binary information. This has not been investigated further as it is not a central aspect of the research. It would be interesting, if not particularly useful, to explore the limits of such an architecture. It is, however, unreasonable to expect that it will provide any great benefits over current information storage methods as my experiments suggest that the pattern of weights for the Hopfield net require as many bits to store or transmit as the original binary message. The storage capacity of a Hopfield net does indeed appear to be of the order of $n$ binary patterns, each of which is $n$ bits in length ($n$ being the number of cells in the network), *i.e.* of the order of $n^2$ bits, but it is a well connected architecture (each cell is connected to all other cells, even if those connections have a weight of zero), so it requires $n^2$ connections. Hence there is no advantage as regards storage capacity to using a Hopfield net.

It might be argued that a network of 3-4 cell assemblies has no more use than the equivalent Hopfield net, and takes a great deal more memory to implement. I do not yet have a satisfactory answer to this point. I can only suggest that the dynamic properties of cell assemblies may lead to such networks having some practical use. In this respect, the temporal nature of cell assemblies, the fact that their activation fades with time, may prove useful. While extra patterns can be stored in a Hopfield net at any point during its use, *i.e.* not only during the initial "training" session, it is not generally used in that manner. It is also a lot harder to remove patterns

from a Hopfield net once it has been stored. Cell assemblies form as a result of Hebbian learning and there is no reason why this should be turned off once assemblies have formed. As a result, cell assemblies can be enlarged, forgotten or split into subassemblies that can be ignited independently. For this reason, I do not dismiss 3-4 networks of cell assemblies as being useless. This is one area that requires further investigation.

### 6.3.2 Practical applications

Simulated cell assemblies have been established as connectionist architecture with some potential for information processing, but without any practical uses they can only ever be a side-line in the connectionist approach to AI. There are several research areas that, although they have yielded good results with connectionist nets such as MLPs and Kohonen SOTMs, have still not been fully solved. A good example is speech recognition [21]. Although standard connectionist architectures have been applied to speech recognition problems [118], they have only achieved high success rates in strictly controlled circumstances (such as isolated word speech and severely restricted vocabularies). This occurs because MLPs generally require the format of inputs to be highly restricted. For example, since a trained MLP has a fixed number of input connections, input patterns representing words must be compressed or expanded to match that number of inputs. Cell assemblies are more flexible in this respect, with assembly ignition occuring simply when the number of firing cells is sufficiently high.

This flexibility may allow cell assemblies to be applied fruitfully to many areas in which standard neural network architectures have failed to fulfil their initial promise. I have already shown that cell assemblies can operate as associative memories, so they can be applied in any area where an associative memory is required, such as image storage and retrieval. Already cell assemblies have been proposed as a practical means of datamining and for use in Internet search engines [92], and as their capabilities are researched further, we can expect other applications for them to be found.

# Bibliography


[1] Abeles, M. (1991) *Corticonics: Neural Circuits of the Cerebral Cortex*, Cambridge University Press, Cambridge, UK.

[2] Abeles, M., Bergman, H, Margalit, E., Vaadia, E, (1993) "Spatiotemporal firing patterns in the frontal cortex of behaving monkeys," *Journey of Neurophysiology*, **70**(4), pp. 1629-38.

[3] Abbott, L.F., Marder, E., (1995), "Activity-dependent regulation of neuronal conductances", *The Handbook of Brain Theory and Neural Networks*, ed. M.A. Arbib, pp. 63-65, MIT Press: Cambridge, MA.

[4] Amit, D.J., (1988) "Neural networks counting chimes," *Proceedings of the National Academy of Sciences, USA*, April 1988, **85**, pp. 2141-2145.

[5] Amit, D.J., (1989) *Modelling Brain Function: The World of Attractor Neural Networks*, Cambridge University Press.

[6] Amit, D. J. (1995) "The Hebbian paradigm reintegrated: Local reverberations as internal representations", *Behavioural Brain Science*, **18**, pp. 617-657.

[7] Anderson, J.A., (1995) *An Introduction to Neural Networks*, MIT Press, Cambridge MA.

[8] Anderson, J.A., Silverman, J.W., Ritz, S.A., Jones, R.S., (1977), "Distinctive features, categorical perception, and probability learning: Some applications of a neural model", *Psychological Review*, **85**, pp. 413-451

[9] Bale, T.A., (1998), *Modular Connectionist Architectures and the Learning of Quantification Skills*, Thesis submitted in partial fulfilment for the requirements

of the degree of Doctor of Philosophy, Department of Computing, University of Surrey, Guildford, UK.

[10] Bear, M.F., Conners, B.W., Paradisa, M.A. (2001), "Neuroscience: Exploring the Brain", Lippincott: Baltimore.

[11] Bechtel, W., Abrahamsen, A., (1991), *Connectionism and the Mind*, Blackwell.

[12] Beurle, R.L., (1954), "Properties of a Mass of Cells Capable of Regenerative Pulses", *Philosophical Transactions Series B* **240**, pp. 55-94.

[13] Bevan, M.D., Wilson, C.J., (1999), "Mechanisms underlying spontaneous oscillation and rhythmic firing in rat subthalamic neurons," *Journal of Neuroscience*, **19**, pp. 7617-7628.

[14] Bi, G.Q., Poo, M.M., (1989), "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type", *Journal of Neuroscience* **18** pp. 10464-10472.

[15] Bi, G.Q., Poo, M.M., (2001), "Synaptic Modification by Correlated Activity: Hebb's Postulate Revisited", *Annual Review of Neuroscience* **24**, pp. 139-166.

[16] Blasdel, G.G., Salama, G. (1986), "Voltage-sensitive dyes reveal a modular organization in monkey striate cortex", *Nature*, **321**, pp. 579-585.

[17] Bliss, T. V. P., Lomo, T. (1973) "Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path." *Journal of Physiology*, **232**, pp. 331-356.

[18] Bloom, L., (1973) *One word at a time*, Mouton: The Hague and Paris.

[19] Bogacz, R., Brown, M.W., Giraud-Carrier, C., "High Capacity Neural Networks for Familiarity Discrimination", *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN99)*, pp. 773-778.

[20] Bostrom, N., (2000), "Cortical Integration", *Consciousness and Cognition* **9(2)**, pp. 39S-40S.

[21] Bowles, R.L., (1992) *Combination of Evidence in Speech Recognition, Thesis Submitted for the Degree of Master of Philosophy*, Department of Electronics and Computer Science, University of Southampton, July 1992.

[22] Bowles, R.L., (2000), "An Introduction to Cellular Assemblies", *Proceedings of CAMDEX 2000 Conference on Good Practice in Computing, 31st March*, Middlesex University, Middlesex, UK.

[23] Braitenberg, V. (1978), "Cell assemblies in the cerebral cortex," in *Theoretical approaches to complex systems*, ed. G. Palm, Springer: Berlin, pp. 171-188.

[24] Braitenberg, V., (1985), "Some Arguments for a Theory of Cell Assemblies in the Cerebral Cortex", *Neural Connections, Mental Computation: Chapter 5* eds. Nadel, Cooper, Culicover, Harnish, MIT Press: Cambridge.

[25] Braitenberg, V., Schuz, A., (1991), *Anatomy of the Cortex: Statistics and Geometry*, Springer-Verlag: Berlin.

[26] Buchanan, B.G., Shortliffe, E.H., (Eds.) (1984), *Rule-Based Expert Systems: The MYCIN Experiments and the Stanford Heuristic Programming Project.* Addison-Wesley: Reading, MA.

[27] Burkitt, A.N., (1996), "Retrieval properties of attractor neural networks that obey Dale's Law using a self-consistent signal-to-noise analysis", *Network: Computing Neural Systems, August 1996*, **7**(3), pp. 517-531.

[28] Brindley, G.S., (1969), "Nerve net models of plausible size that perform many simple learning tasks", *Proceedings of the Royal Society (London), B*, **174**, pp. 173-191.umns, Modules, and Hebbian Cell Assemblies", *The Handbook of Brain Theory and Neural Networks*, Bradford Books/MIT Press, pp. 269-272.

[29] Calvin, W.H., (1995), "Cortical Columns, Modules, and Hebbian Cell Assemblies", *Handbook of Brain Theory and Neural Networks*, ed. Arbib, M.A., MIT/Bradford Books, Cambridge, MA, pp. 269-272.

[30] Chalmers, D.J., (1990), "Why Fodor and Pylyshyn Were Wrong: The Simplest Refutation", *Proceedings of The Twelfth Annual Conference of the Cognitive*

*Science Society, Cambridge, MA, July 1990*, Lawrence Erlbaum Associates: Hillsdale, NJ, pp. 340-347.

[31] Chalup, S.K., (2002), "Incremental Learning in Biological and Machine Learning Systems", *International Journal of Neural Systems* **12(6)** pp. 447-465.

[32] Cheeseman, P., (1985), "In Defense of Probability", *9th International Joint Conference on AI*, pp. 1002-1009.

[33] Chiueh, T.D., Goodman, R.M., (1991), "Recurrent Correlation Associative Memories", *IEEE Transactions on Neural Networks*, **2**, pp. 275-284.

[34] Choe, Y., Miikkulainen, R., (2003), "The role of postsynaptic potential decay rate in neural synchrony", *Neurocomputing* **52-54**, pp. 707-712.

[35] Chown, E. (1994). *Consolidation and learning: A connectionist model of human credit assignment.* Ph.D. Thesis, University of Michigan, USA.

[36] Churchland, P.S., T.J. Sejnowski (1994) *The Computational Brain*, MIT Press.

[37] Damasio, A.R., (1989), "Time-locked multiregional retroactivation: A systems-level proposal for the neural substrates of recall and recognition", *Cognition* **33**, pp. 25-62.

[38] Davey, N., Adams, R., Hunt, S., (2000), "High Performance Associative Memory Models and Symmetric Connections", *Proceedings of the International ICSC Congress on Intelligent Systems and Applications (ISA 2000): Symposium on Computational Intelligence (CI 2000)* **2**, pp. 326-331, Wollongong, Australia, December 2000.

[39] Davey, N., Hunt, S.P., (2000), "A Comparative Analysis of High Performance Associative Memory Models", *Proceedings of 2nd International ICSC Symposium on Neural Computation (NC'2000)*, pp. 55-61.

[40] Davidson, P.R., (2001), "Computational Modelling of the Human Motor Control System: Nonlinear Enhancement of the Adaptive Model Theory through Simulation and Experiment", Ph.D. thesis, University of Canterbury, U.K.

[41] Dayan, P., Abbott, L.F., (2001), *Theoretical Neuroscience. Computational and Mathematical Modeling of Neural Systems*, MIT Press: Cambridge.

[42] Diederich, S., Opper, M., (1987), "Learning of Correlated Patterns in Spin-Glass Networks by Local Learning Rules", *Physical Review Letters* **58**, pp. 949-952.

[43] Donoghue, J.P., Sanes, J.N., Hatsopoulos, N.G., Gaal, G. (1998), "Neural discharge and local field potential oscillations in primate motor cortex during voluntary movements", *Journal of Neurophysiology* **79**, pp. 159-73.

[44] Duda, R.O., Hart, P.E., Nilsson, N.J., (1990), "Subjective Bayesian Methods for Rule-Based Inference Systems", *Readings in uncertain reasoning*, Morgan Kaufman: San Francisco, pp. 274-281.

[45] Eccles, J.C., (1964), *The Physiology of Synapses*, Springer-Verlag: New York

[46] Eckhorn, R., (1999), "Neural mechanisms from the visual cortex suggest basic circuits", *Transactions on Neural Networks 10*, pp. 464 - 479.

[47] The Economist, (2005), *Grey matter, blue matter*, June 9th, 2005.

[48] Edelman, G.M., (1992), *Bright air, brilliant fire: on the matter of the mind*, Basic Books: New York.

[49] Edelman, G.M., (1998), *Building a picture of the brain*, Daedalus, Spring 1998, pp. 37-70.

[50] Ermentrout, G.B, Cowan, J.D, (1979), "A mathematical theory of visual hallucination patterns.", *Biological Cybernetics* **34(3)**, pp. 137-150.

[51] Feller, W., (1968), *An Introduction to Probability Theory and Its Applications. vol. I*, Wiley: New York.

[52] Fodor, J.A., Pylyshyn, Z.W., (1988), "Connectionism and cognitive architecture: A critical analysis", *Cognition* **28**, pp. 3-71.

[53] Forbell, E., Chown, E, (2000), "Lexical contact during speech perception: A connectionist model", *Proceedings of the Twenty Second Annual Meeting of the Cognitive Science Society*, pp. 142-147.

[54] Fransen, E., Lansner, A., Liljenstr"om, H., (1993), "A Model of Cortical Associative Memory Based on Hebbian Cell Assemblies", *Computation and Neural Systems*, ed. Eeckman, Bower, pp. 431-435, Kluwer Academic: Amsterdam.

[55] Freiwald, W.A., Kreiter, A.K., Singer, W., (2001), "Synchronization and assembly formation in the visual cortex", *Progress in Brain Research: Chapter 8* **130**, Elsevier Science.

[56] Friedman, L., (1981), "Extended Plausible Inference", *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pp. 487-495.

[57] Gage, N., Hickok, G., (2005), "Multiregional cell assemblies, temporal binding, and the representation of conceptual knowledge in cortex: A modern theory by a "classical neurologist, Carl Wernicke", submitted for publication in *Cortex*.

[58] Gardner, E., (1987), "Maximum Storage Capacity in Neural Networks", *Europhysics Letters* **4(4)**, pp. 481-485.

[59] Gardner, E., (1988), "The space of interactions in neural network models", *Journal of Physics, A*, **21**, pp. 257-270.

[60] Gerstein, G., Perkel, D., Dayhoff, J. (1985), "Cooperative .ring activity in simultaneously recorded populationsof neurons: Detection and measurement", *Journal of Neuroscience* **5**, pp. 881889.

[61] Gilbert C.D., Hirsch J.A., Wiesel T.N., (1990) "Lateral interactions in visual cortex", *Cold Spring Harbor Symposia on Quantitative Biology* **LV**, Cold Spring Harbor Laboratory Press. pp. 663-676.

[62] Goertzel, B., (1994), "Periodic Brain Responses and Beyond", *Psycoloquy* **5(51)** Brain Rhythms (3).

[63] Goertzel, B., (1996), *From Complexity to Creativity: Computational Models of Evolutionary, Autopoietic and Cognitive Dynamics*, Chapter 10, Plenum Press: New York.

[64] Graham, B, Willshaw, D, (1997), "Capacity and information efficiency of the associative net," *Network* **8**, pp. 35-54.

[65] Gray, C.M., König, P., Engel, A.K., Singer, W., (1989) "Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties", *Nature* **388**, pp. 334337.

[66] Grossberg, S., (1974), *Classical and instrumental learning by neural networks* in Progress in theoretical biology, R. Rosen and F. Snell (Eds.), Academic Press: New York.

[67] Grun, S., (1996), "Unitary joint-events in multiple-neuron spiking activity-detection, significance and interpretation", Verlag Harry Deutsch: Frankfurt.

[68] Gulyas, A.I., Toth, K., McBain, C.J., Freund, T.F., (1998), "Stratum radiatum giant cells: a type of principal cell in the rat hippocampus", *European Journal of Neuroscience* **10(12)**, December 1998, p. 3813-3822.

[69] Hancock, E.R., Pelillo M., (1998), "A Bayesian Interpretation for the Exponential Correlation Associative Memory", *Pattern Recognition Letters*, **19**(2), pp. 149-159.

[70] Hanganu, I.L., Kilb, W., Luhmann, H.J., (2001), "Spontaneous Synaptic Activity of Subplate Neurons in Neonatal Rat Somatosensory Cortex", *Cerebral Cortex* **11(5)**, pp. 400-410, Oxford University Press.

[71] Hanson, S., (1999), "Emergent Properties of Neural Networks in Language and Event Parsing Tasks: Implications for Neuroscience and Neuroimaging Data", *Proceedings of EmerNet, International Workshop on Current Computational Architectures Integrating Neural Networks and Neuroscience, 8th-9th August 2000, Durham Castle, Durham, United Kingdom.*

[72] Harston, C. T., (1990), "Learning background for neural networks", in *Handbook of neural computing applications*, Maren, A.J., Hartson, C.T., Pap, R.M. (eds), p. 451, Academic Press: San Diego.

[73] Haykin, S., (1999), *Neural Networks. A Comprehensive Foundation*, 2nd Edition, Prentice-Hall Inc.

[74] Hebb, D.O., (1949), *Organization of Behavior*, John Wiley & Sons: New York.

[75] Hebb, D.O. (1958), *A Textbook of Psychology* W. B. Saunders Company: Philadelphia.

[76] Hebb, D.O., (1980), *Essay on Mind*, Erlbaum: Hillsdale, New Jersey

[77] Hertz, J., Krogh, A., Palmer, R.G., (1991), *Introduction to the Theory of Neural Computation*, Perseus Books: Cambridge, MA.

[78] Hetherington, P.A., Shapiro, M., (1993), "Simulating Hebb cell assemblies: the necessity for partitioned dendritic trees and a post-not-pre LTD rule", *Network: Computations in Neural Systems* **4**, pp. 135-153.

[79] Hjorth, J., (2003), "Chunking of Action Sequences in the Cortex-Basal Ganglia System", Masters Thesis in Computer Science, School of Engineering Physics, Royal School of Technology, Sweden.

[80] Hodgkin, A.L., Huxley, A.F., (1952), "A quantitative description of membrane current and application to conduction and excitation in nerve", *Journal of Physiology* **117**, pp. 500-544.

[81] Hoffmann, A., (1998), *Paradigms of Artificial Intelligence: A Methodological and Computational Analysis*, Springer-Verlag: Berlin.(1992), *Adaptation in natural and artificial systems*, MIT Press: MA.

[82] Holland, J., (1992), "Genetic algorithms", *Scientific American* **267(1)**, pp.66-72.

[83] Holland, J.H., (1998), *Emergence: From Chaos to Order*, Perseus Books (January 1998).

[84] Hopfield, J.J., (1982), "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proceedings of the National Academy of Sciences, USA (April)* **79**, pp. 2554-2558.

[85] Hopfield, J.J., Tank, D.W., (1986), "Computing with Neural Circuits: A Model", *Science (August 1986)* **233**, pp. 625-633.

[86] Hubel, D.H., Wiesel, T.N., (1968), "Receptive fields and functional architecture of the monkey striate cortex", *Journal of Physiology, London* **195**, pp. 215-243.

[87] Hubel, D.H., Wiesel, T.N., (1974), "Sequence regularity and geometry of orientation columns in the monkey striate cortex", *Journal of Comparative Neurology* **158**, pp. 267-294.

[88] Hubel, D.H., Wiesel, T.N., (1977), "Functional architecture of macaque monkey visual cortex", *Proceedings of the Royal Society (B)* **198**, pp. 1-59.

[89] Huyck, C.R., (2000), "Modelling Cell Assemblies", *Proceedings of the International Conference on Artificial Intelligence*, pp. 891-897.

[90] Huyck, C.R., (2001), "Cell Assemblies as an Intermediate Level Model of Cognition", *Emergent neural computational architectures based on neuroscience: towards neuroscience-inspired computing*, pp. 383-397, Springer-Verlag: New York.

[91] Huyck, C.R., (2002), "Cell Assemblies and Neural Network Theory: From Correlators to Cell Assemblies", *Middlesex University Technical Report ISSN 1462-0871 CS-02-02.*

[92] Huyck, C.R., Bavan, S., (2002), "Neural Cell Assemblies for Practical Applications", 6th World Multiconference on Systemics, Cybernetic, and Informatics (SCI2002), July 14-18, 2002, Sheraton World Resort, Orlando, Florida, USA.

[93] Huyck, C.R., Bowles, R.L., (2004). "Spontaneous neural firing in biological and artificial neural systems". *Journal of Cognitive Systems* **6**(1), pp. 31-40.

[94] Huyck, C.R., Orengo, V., (2005), "Information Retrieval and Categorisation using a Cell Assembly Network", *Neural Computing and Applications* **14**, pp. 282-289.

[95] Iglesias J., Eriksson J., Pardo B., Tomassini M., Villa A.E.P. (2005), "Emergence of Oriented Cell Assemblies Associated with Spike-Timing-Dependent Plasticity", *ICANN 2005, Lecture Notes in Computer Science, LNCS 3696*, pp. 127-32.

[96] Izhikevich, E.M, (2004), "Which Model to Use for Cortical Spiking Neurons?", *IEEE Transactions on Neural Networks, September 2004* **15**(5), pp. 1063-1070.

[97] Jacobs, R.A., Jordan, M.I., Barto, A.G., (1991), "Task Decomposition Through Competition in a Modular Connectionist Architecture: The What and Where Vision Tasks", *Cognitive Science* **15**, pp. 219-250.

[98] Kaplan, S., Sonntag, M., Chown, E., (1991), "Tracing Recurrent Activity in Cognitive Elements (TRACE): a Model of Temporal Dynamics in a Cell Assembly", *Connection Science* **3**(2), pp. 179-206.

[99] Katz L.C., Callaway E.M., (1992), "Development of local circuits in mammalian visual cortex", *Annual Review Neuroscience* **15**, pp. 31-56.

[100] Kelso, S.R., Ganong, A.H., Brown, T.H., (1986), "Hebbian synapses in hippocampus", *Proceedings of the National Academy of Sciences, USA* **83** pp. 53265330.

[101] Kitano, K., Aoyagi, T., (1989), "Retrieval dynamics of neural networks for sparsely coded sequential patterns", *Journal of Physics A: Math. General* **31(36)** pp. L613-L620.

[102] Knoblauch, A., Markert, H., Palm, G., (2004), "An associative model of cortical language and action processing", *Proceedings of NCPW9, Neural Computation and Psychology Workshop*, University of Plymouth.

[103] Knoblauch, A., Fay, F., Kaufmann, U., Markert, H., Palm, G., (2004), "Associating words to visually recognized objects", *Anchoring symbols to sensor data. Papers from the AAAI Workshop. Technical Report WS-04-03* eds. S.Coradeschi, S., Saffiotti, A., pp 10-16, AAAI Press, Menlo Park, California.

[104] Kohonen, T., (1977), *Associative memory; A system theoretic approach*, Springer: Berlin.

[105] Kohonen, T., (1984), *Self-Organization and Associative Memory*, Springer-Verlag: Berlin.

[106] Konorski, J., (1967), *Integrative Activity of the Brain*, University of Chicago, US.

[107] Kosko, B., (1986) "Differential Hebbian Learning" in *Neural Networks for Computing: AIP Conference Proceedings* ed. J.S. Denker, pp. 265-270, Americal Institute of Physics: New York.

[108] Krauth, W., Opper, M., (1989), "Critical storage capacity of the j = +1 neural network", *Journal of Physics A* **22**, pp. 519-523.

[109] Landauer, T., (1986), "How Much Do People Remember? Some Estimates of the Quantity of Learned Information in Long-term Memory", *Cognitive Science* **10**, pp. 477-493.

[110] Lansner, A., (1982), "Information processing in a network of model neurons. A computer simulation study", *Technical Report TRITA-NA-8211*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm

[111] Lansner, A., Fransen, E., (1992), "Modelling Hebbian cell assemblies comprised of cortical neurons", *Network* **3**, pp. 105-119.

[112] Larson, D., Lynch, G., (1989), "Theta pattern stimulation and the induction of LTP: the sequence in which synapses are stimulated determines the degree to which they potentiate", *Brain Research* **489** pp. 4958.

[113] Lashley, K.S., (1951), *The problem of serial order in behavior* in Cerebral mechanisms of behavior; The Hixon Symposium, L.P. Jeffress (Ed.), Wiley: New York.

[114] LeVay, S., Stryker, M.P., Shatz, C.J., (1978), "Ocular dominance columns and their development in layer IV of the cat's visual cortex: A quantitative study." *Journal of Comparative Neurology*, **179**, pp. 223-244.

[115] Levy, N., Horn, D., Ruppin, E., (1999), "Associative memory in a multimodular network", **Neural Computation 11**, pp. 17171737.

[116] Lindley, D.V., (1987), "The Probability Approach to the Treatment of Uncertainty in Artificial Intelligence and Expert Systems", *Statistical Science* **2(1)**, pp. 3-44.

[117] Lippmann, R.P., (1987), "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine, April 1987*, pp. 4-22.

[118] Lippmann, R.P., (1990), "Review of Neural Networks for Speech Recognition", *Neural Computation* **1(1)**, pp. 1-38, MIT Press: Cambridge, MA.

[119] MacGregor, R., McMullen, T., (1978), "Computer simulations of diffusely connected neuronal populations", *Biological Cybernetics* **28**, pp. 121-127.

[120] MacKay, D.J.C,, (1997), "Information Theory, Pattern Recognition and Neural Networks", Chapter 15, pp. 175-184, Cambridge University Press.

[121] Marr, D., (1969), "A theory of cerebellar cortex", *Journal of Physiology* **202**, pp. 437-470.

[122] Marr, D., (1971), "Simple memory: A theory for archicortex", *Philosophical Transactions of the Royal Society of London B* **262**, pp. 23-81.

[123] Martignon, L., Deco, G., Laskey, K., Diamond, M., Freiwald, W., Vaadia, E., (2000), "Neural Coding: Higher-Order Temporal Patterns in the Neurostatistics of Cell Assemblies", *Neural Computation* **12**, pp. 26212653.

[124] Martin, N., Dell, G.S., Saffran, E.M., Schwartz, M.F., (1994), "Origins of Paraphasias in Deep Dysphasia: Testing the Consequences of a Decay Impairment to an Interactive Spreading Activation Model of Lexical Retrieval", *Brain and Language* **47**, pp. 609-660.

[125] Martin, N., Saffran, E.M., Dell, G.S, (1996), "Recovery in Deep Dysphasia: Evidence for a Relation between Auditory-Verbal STM Capacity and Lexical Errors in Repetition", *Brain and Language* **52**, pp. 83-113.

[126] McClelland, J.L., (1981), "Retrieving general and specific information from stored knowledge of specifics", *Proceedings of the Third Annual Conference of the Cognitive Science Society*, pp. 170-172.

[127] McClelland, J. L., (1981), "Retrieving general and specific information from stored knowledge of specifics", *Proceedings of the Third Annual Conference of the Cognitive Science Society*, pp. 170-172.

[128] McCullogh, W. S., Pitts, W., (1943). "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics* **5**, pp. 115-133.

[129] McGregor, R.J. (1993) "Composite cortical networks of multimodal oscillators", *Biological Cybernetics* **69**, pp. 243-255.

[130] Mesulam, M.M., (1998) "From sensation to cognition", *Brain* **121**, pp. 1013-1052.

[131] Milde, G., Kobe, S., (1997), "An exact learning algorithm for autoassociative neural networks with binary couplings", *Journal of Physics A: Mathematics General* **30**, pp. 2349-2352.

[132] Miller, R., (1996), "Neural assemblies and laminar interactions in the cerebral cortex", *Biological Cybernetic* **75**, pp. 253-261.

[133] Milner, P.M. (1974) "A model for visual shape recognition", *Psychological Review* **81(6)**, pp. 521535.

[134] Milner, P.M., (1996), "Neural Representations: Some Old Problems Revisited", *Journal of Cognitive Neuroscience* **8**, pp. 69-77.

[135] Milner, P.M., (1999), *The Autonomous Brain*, Erlbaum: New Jersey.

[136] Minsky, M. A., Papert, S. (1969) *Perceptrons*, Cambridge, MA: MIT Press.

[137] Nicolelis, M., De Schutter, E., (2001) "Special Issue on Multiple Electrode Recordings", Journal of Neuroscience Methods" **94** no. 1, pp. 3-154.

[138] Palm, G., (1982), *Neural Assemblies, An Alternative Approach to Artificial Intelligence*, Springer-Verlag: Berlin.

[139] Palm, G., (1986), "On the internal structure of cell assemblies," *Brain Theory*, eds. Palm, G., Aertsen, A., Springer: Berlin, Heidelberg: New York, Tokyo, p. 261.

[140] Peelle, J.E., Vogels, T., Abbott, L.F., (2005), "Temporal sequences in sparsely-connected networks of integrate-and-fire neurons", *Annual Meeting of the Society for Neuroscience*, Washington, DC, November 2005.

[141] Peng, Y, Zhou, Z, (1996), "Turning Backpropagation Networks into High-Capacity Associative Memories, *Proceedings of the World Congress on Neural Networks*, San Diego, CA, September, 1996, pp. 743-748.

[142] Posner, M.I., Raichle, M.E., (1994), *Images of Mind*, Scientific American Library: New York.

[143] Pribram, K.H., (1991), '*Brain and Perception*, Lawrence Erlbaum Associates: Hillsdale, New Jersey.

[144] Prut, Y., Vaadia, E., Bergman, H., Haalman, I., Slovin, H., Abeles, M, (1998), "Spatiotemporal structure of cortical activity: Properties and behavioral relevance", *Journal of Neurophysiology* **79**, pp. 28572874.

[145] Pulvermüller, F., (1992), "Constituents of a neurological theory of language", *Concepts in Neuroscience* **3**, pp. 157-200.

[146] Pulvermüller, F., (1999), "Words in the Brain's Language", *Behavioral and Brain Sciences* **22** pp. 253-336.

[147] Ramon y Cajal, S., (1928), *Degeneration and Regeneration of the Nervous System*, Oxford University Press: London.

[148] Riehle, A., Grün, S., Diesmann, M., Aertsen, A., (1997) "Spike Synchronization and Rate Modulation Differentially Involved in Motor Cortical Function", *Science* **278**, pp. 573-578

[149] Rizzolatti, G., Arbib, M.A., (1998) "Language within our grasp", *Trends in Neuroscience*, **21** pp. 188-194.

[150] Rochester, N., Holland, J., Haibt, L., Duda, W., (1956), "Tests on a cell assembly theory of the action of the brain using a large digital computer", *IRE Transactions on Information Theory* (IT-2), pp. 80-93.

[151] Rosenblatt, F., (1959). "Two theorems of statistical separability in the perceptron." *Mechanization of Thought Processes*, **1**, London: HMSO.

[152] Rumelhart, D. E., McClelland, J. L., and the PDP Research Group (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, **1**: *Foundations*, Cambridge, MA: MIT Press/Bradford Books.

[153] Sakurai, Y., (1996), "Population coding by cell assemblies - what it really is in the brain", *Neuroscience Research* **26(1)**, pp. 1-16.

[154] Sakurai, Y., (1998), "The search for cell assemblies in the working brain", *Behavioural Brain Research* **91**, pp. 1-13.

[155] Sholl, D.A., (1956), *The Organisation of the Cerebral Cortex*, Wiley: New York.

[156] Segev, I., (1995), "Dendritic Processing", in *The Handbook of Brain Theory and Neural Networks*, M.A. Arbib (Ed.), MIT/Bradford Books: Cambridge, MA, pp. 282-286.

[157] Sejnowski, T.J., Rosenberg, C.R., (1987), "Parallel Network that Learn to Pronounce English Text, *Complex Systems* **1(1)**, pp. 145-168.

[158] Sejnowski, T.J., Koch, C., Churchland, P.S., (1989), "Computational neuroscience", *Science* **241**, pp. 1299-1306.

[159] Selfridge, O. G. (1959) "Pandemonium: A paradigm for learning", *Symposium on the Mechanization of Thought Processes*, London: HMSO.

[160] Shafer, G., (1987) "Probability Judgment in Artificial Intelligence and Expert Systems", *Statistical Science* **2(1)**, pp. 3-44.

[161] Shannon, C.E., (1948), "A Mathematical Theory of Communication", *Bell System Technical Journal,July/October* **27**, pp. 379-423.

[162] Shannon, C.E., Weaver, W., (1963), "The Mathematical Theory of Communication", University of Illinois Press.

[163] Sharkey, A.J.C., (1996), "On Combining Artificial Neural Nets", *Connection Science*, **8**, pp. 299-314.

[164] Shaw, G.L., Silverman, D.J., Pearson, J.C., (1985) "Model of cortical organization embodying a basis for the theory of information processing and memory recall", *Proceedings of the National Academy of Science*, **82**, pp. 2364-2368.

[165] Sikstrom, S.P., (1996), "The TECO Connectionist Theory of Recognition Failure", *European Journal of Cognitive Psychology* **8(4)**, December 1996, Psychology Press, pp. 341-380.

[166] Singer, W., (1990), "The formation of cooperative cell assemblies in the visual cortex", *Journal of Experimental Biology* **153** (October 1990), pp. 177-197.

[167] Singer, W., (1995), "Development and Plasticity of Cortical Architectures," *Science*, **270**, pp. 758-763.

[168] Singer, W., Gray, C. M., (1995) "Visual feature integration and the temporal correlation hypotheses", *Annual Review of Neuroscience* **8**, pp. 555 - 568.

[169] Singer, W., (1998), "Consciousness and the structure of neuronal representations", *Philosophical Transactions of the Royal Society of London, B, Biological Sciences, Nov 1998*, **29;353(1377)**, pp. 1829-1840.

[170] Smith, S.L., Otis, T.S., (2003), "Persistent Changes in Spontaneous Firing of Purkinje Neurons Triggered by the Nitric Oxide Signaling Cascade", *The Journal of Neuroscience* **23(2)**, January 2003, pp. 367-372.

[171] Smolensky, P., (1987), "The constituent structure of connectionist mental states: A reply to Fodor and Pylyshyn", *Southern Journal of Philosophy* **26**, pp. 137-163.

[172] Smolensky, P., (1987), "Connectionist AI, symbolic AI, and the brain", *Artificial Intelligence Review* **1**, pp. 95-109.

[173] Sohn, J-W., Averbeck, B.B., Lee, D., (2002), "Temporal precision in the transmission of information between neurons in the primate supplementary motor area", *Society for Neuroscience, Abstracts* **28**.

[174] Sommer, F.T., Wennekers, T., Palm, G., (1998), "Bidirectional completion of cell assemblies in the cortex", *Computational Neuroscience: Trends in Research 1998*, Plenum Press: New York.

[175] Sonntag, M., (1991), *Learning sequences in an associative network: A step towards cognitive structure*, Ph.D. Thesis, University of Michigan, USA.

[176] Spivey, M., Andrews, M., Richardson, D., (1999) "On computational and behavioral evidence regarding Hebbian transcortical cell assemblies", *Behavioral and Brain Sciences* **22:2**, p. 302.

[177] Spratling, M.W., "Cortical region interactions and the functional role of apical dendrites", *Behavioral and Cognitive Neuroscience Reviews* **1(3)**, pp. 219228.

[178] Stanton, P.K., Sejnowski, T.J., (1989), "Associative long-term depression in the hippocampus induced by Hebbian covariance", *Nature* **339**, pp. 215-217.

[179] Sutton, R.S., Barto, A.G., (1988), "Toward a Modern Theory of Adaptive Networks: Expectation and Prediction", *Psychological Review* **2**, pp. 135-170.

[180] Swerdlow, J.L., "Quiet Miracles of the Brain", *National Geographic, June 1995*, pp. 2-41.

[181] Tansey, E.M., (1991), "Chemical neurotransmission in the autonomic nervous system: Sir Henry Dale and acetylcholine", *Clinical Autonomic Research* **1**(1), pp. 63-72.

[182] Tanzi, E., (1893), "I fatti e le induzioni nell' odierna isologia del sistema nervoso", *Rivista Sperimentale di Freniatria e Medicina legale* **19**, pp. 419-472.

[183] Thomas, W.V., Alexander, I., Bowden, P.A., (1984), "Wisard: a radical step forward in image recognition", *Sensor Review* **4**, pp. 120-124.

[184] Vander, A., Sherman, J., Luciano, D., (1998), *Human Physiology. The Mechanism of Body Function*, 7th Edition, McGraw-Hill.

[185] von der Malsburg, C., (1973), "Self-organization of orientation sensitive cells in the striate cortex", *Kybernetic* **14**, pp. 85-100.

[186] Wickelgren, W.A., (1992), "Webs, cell assemblies, and chunking in neural nets", *Concepts in Neuroscience*, **3**, pp. 1-53.

[187] Williams, R.W., Herrup, K., (2001) "The control of neuron number", *The Annual Review of Neuroscience* **11**, pp. 423-453.

[188] Wilson, C.J., Groves, P.M., (1981), "Spontaneous firing patterns of identified spiny neurons in the rat neostriatum", *Brain Research* **220(1)** September 1981, pp. 67-80.

[189] Wilson, R.C., Hancock, E.R., (1999), "Storage Capacity of the Exponential Correlation Associative Memory", *Lecture Notes In Computer Science; Vol. 1606, Proceedings of the International Work-Conference on Artificial and Natural Neural Networks: Foundations and Tools for Neural Modeling*, pp. 301-310, Springer-Verlag.

[190] Wolfe, J.M., (1994), "Guided Search 2.0. A Revised Model of Visual Search", *Psychonomic Bulletin and Review* **1(2)**, pp. 202-238.

[191] Wolff, J.G., (2001), "Neural Mechanisms for Information Compression by Multiple Alignment, Unification and Search". *Technical Report*, School of Informatics, University of Wales at Bangor.

[192] Wolff, J.G., (2003), "Information Compression by Multiple Alignment, Unification and Search as a Unifying Principle in Computing and Cognition", *Artificial Intelligence Review* **19**, 193-230.

[193] Wright, J., Ahmad, K., "The Connectionist Simulation of Aphasic Naming", Technical Report, Artificial Intelligence Group, Department of Mathematical and Computer Science, University of Surrey, Guildford, UK.

[194] Xie, X., Berger, T.W., Barrioneuvo, G., (1991), "Isolated NMDA receptor-mediated synaptic responses in the hippocampus can express both LTP and LTD", *Society for Neuroscience Meeting Abstract no 161.8*

[195] Zadeh, L.A., (1984), "Review of a mathematical theory of evidence," *AI Magazine*, **5(3)**, pp. 81-83.

[196] Zadeh, L.A., (1986) "A Simple View of the Dempster-Shafer Theory of Evidence and its Implication for the Rule of Combination", *AI Magazine* **7(2)**, Summer 1986, pp. 85-90.

[197] Zeki, S., (1993), *A Vision of the Brain*, Blackwell Science: Oxford.

# Appendix A

# Derivation of Some of the Properties of the Hopfield Net

This appendix summarises the proof of the storage capacity given in [77]. Define the following terms:

| | |
|---|---|
| $N$ | The number of cells in the Hopfield net |
| $p$ | The number of patterns stored in the net |
| $w_{ij}$ | The strength of the weight from cell $i$ to cell $j$ |
| $\mu_i(t)$ | The output of cell $i$ at time $t$ |
| $\mu_i(0)$ | Bit $i$ of the test pattern, presented to cell $i$ at time $t = 0$ |
| $h_i$ | The sum of the inputs $\mu_i$ weighted by connection strengths<br>$h_i = \sum_j^N w_{ij}\mu_j(t)$ |
| $\xi_i^s$ | Bit $i$ of pattern $s$ on which the Hopfield net is to be trained |
| $\xi_i$ | Bit $i$ in cases in which a single training pattern is used |
| $f_k$ | Hard-limiting ("sgn") function such that<br>$f_k(x)$ is +1 for $x \geq 0$, −1 for $x < 0$ |

The weights of the Hopfield net should be set up in such a way that the training patterns themselves are stable, *i.e.* that any given bit does not change on iteration when presented to the net. This implies that $f_k(h_i) = \mu_i(0)$ for all $i$, provided that $\mu_i(0) = \xi_i^s$, *i.e.* that passing the sum of the weighted inputs for any bit $i$ of the test pattern produces a value with the same sign.

In cases in which there is only a single training pattern to be stored, $\xi_i$ (for $i = 1$

$\ldots N$), the required condition is that

$$sgn(\sum_{j}^{N} w_{ij}\xi_j(t)) = \xi_i \tag{A.1}$$

Take $w_{ij}$ to be proportional to the product of $\xi_i$ and $\xi_j$ ($k$ is the constant of proportionality):

$$w_{ij} = k.\xi_i.\xi_j \tag{A.2}$$

$$\sum_{j}^{N} w_{ij}.\xi_j = \sum_{j}^{N} k.\xi_i.\xi_i.\xi_j = k\sum_{j}^{N} \xi_i.\xi_j.\xi_j \tag{A.3}$$

for $x_j = \pm 1$ $\xi_j.\xi_j = 1$, giving

$$\sum_{j}^{N} w_{ij}.\xi_j = k\sum_{j}^{N} \xi_i = kN\xi_i \tag{A.4}$$

The sign of $kN\xi_i$ must be the same as that of $\xi_i$, so the condition in equation A.1 is guaranteed. $k$ is taken to be $\frac{1}{N}$ for convenience. If the single pattern is replaced by $M$ patterns, then any particular pattern $S$ is stable for all bits $i$ if

$$h_i = \sum_{j=1}^{N} \xi_j^S.w_{ij} \tag{A.5}$$

$$sgn(h_i) = \xi_i^S \tag{A.6}$$

*i.e.* there is no change in any bit when pattern $x^S$ is applied to the input. If the weights are defined using equation A.7,

$$w_{ij} = \frac{1}{N}\sum_{s=1}^{M} \xi_i^s.\xi_j^s \tag{A.7}$$

then

$$\begin{aligned} h_i &= \sum_{j=1}^{N} \xi_j^S(\frac{1}{N}\sum_{s=1}^{M} \xi_i^s.\xi_j^s) \\ &= \frac{1}{N}\sum_{j=1}^{N}\sum_{s=1}^{M} \xi_i^s.\xi_j^s.\xi_j^S \end{aligned} \tag{A.8}$$

This expression can be split into two terms, the first corresponding to $s = S$, and the second corresponding to $s \neq S$. The second term is often called the *crosstalk* term as it corresponds to learned bits in patterns other than $S$ which may differ from those in $S$.

$$h_i = \frac{1}{N}\sum_{j=1}^{N} \xi_i^S . \xi_j^S . \xi_j^S + \frac{1}{N}\sum_{j=1}^{N}\sum_{s\neq S}^{M} \xi_i^s . \xi_j^s . \xi_j^S \tag{A.9}$$

Since $x_j^S = \pm 1$, $\xi_j^S . \xi_j^S = 1$, so the expression reduces to

$$h_i = \frac{1}{N}\sum_{j=1}^{N} \xi_i^S + \frac{1}{N}\sum_{j=1}^{N}\sum_{s\neq S}^{M} \xi_i^s . \xi_j^s . \xi_j^S \tag{A.10}$$

$\xi_i^S$ is a loop invariant in the first term, so $\sum_{j=1}^{N} \xi_i^S$ reduces to $N.\xi_i^S$

$$\begin{aligned} h_i &= \frac{1}{N}(N.\xi_i^S) + \frac{1}{N}\sum_{j}^{N}\sum_{s\neq S}^{M} \xi_i^s . \xi_j^s . \xi_j^S \\ &= \xi_i^S + \frac{1}{N}\sum_{j}^{N}\sum_{s\neq S}^{M} \xi_i^s . \xi_j^s . \xi_j^S \end{aligned} \tag{A.11}$$

Clearly, if the crosstalk term were zero, then equation A.5 would be automatically fulfilled. However, since the equation for updating the bit values only uses the sign of $h_i$, then equation A.5 is fulfilled if the magnitude of the crosstalk term is less than 1, in which case it is incapable of changing the sign of $h_i$. This does in fact turn out to be the case if the number of patterns stored is much smaller than the number of cells. In this case, any of the training patterns presented to the inputs of the Hopfield net will remain unchanged. Furthermore, if a small number of bits in the patterns are wrong on presentation, the updating equation will correct them automatically, so the training patterns themselves form the lowest points of attractor basins.

Define $C_i^S$ to be $\xi_i^S$ times the crosstalk term:

$$C_i^S = \frac{\xi_i^S}{N}\sum_{j=1}^{N}\sum_{s\neq S}^{M} \xi_i^s . \xi_j^s . \xi_j^S \tag{A.12}$$

If $C_i^S$ is greater than -1 for every bit of every pattern, then all the patterns are stable. In that case, either the cross-talk term has the same sign as $\xi_i^S$ or, although

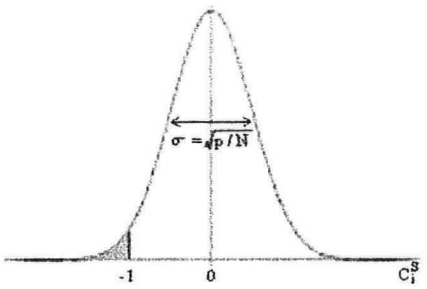


Figure A.1: $C_i^S$ follows an approximately normal distribution.

opposite in sign, unable to alter the sign of $h_i$. For any given bit, we can calculate the probability $p_{err}$ that the bit is not stable

$$p_{err} = P(C_i^S \leq -1) \tag{A.13}$$

Since all patterns consist only of equally likely $\pm 1$ bits, $\xi_i^s . \xi_j^s . \xi_j^S$ must be $\pm 1$. If we assume that both $p$ is much larger than 1 then the double summation in equation A.12 means that $C_i^S$ is approximately equal to $\frac{1}{N}$ times the sum of $Np$ random numbers, each of which is $\pm 1$. The $\xi_i^S$ term in the numerator of equation A.12 makes no difference to this conclusion, since it can only alter the sign of $C_i^S$ not its magnitude. According to the theory of random coin tosses [51, 165], $C_i^S$ must follow a binomial distribution. Providing that $N$ is also large compared to 1, this can be modelled using a normal distribution with mean value 0 and standard deviation $\sqrt{\frac{p}{N}}$, as shown in figure A.1.

The area under the graph is 1 unit, so the shaded area in figure A.1 represents the probability that $C_i^S < -1$. The equation of the normal curve is difficult to process, but the areas under it may be calculated from the standard normal curve tables given at the back of any statistics text book. In this case, we may choose a specific figure for the probability and work backward to find a value for $\frac{p}{N}$. The value of $p_{err}$ is usually taken to be 0.01, giving $\frac{p}{N} = 0.185$, a critical load of 0.185, and shows that the capacity of the network is proportional to the number of cells in the net. This is the probability that any particular bit will change on the first iteration, but it does not preclude the possibility that this will lead the net into a state which is then unstable. In fact, it can be shown that the critical load is 0.138.

# Appendix B

# Significance of the Correlation Coefficient

The *correlation coefficient*, $R$[1], more properly the *Pearson product-moment correlation coefficient*, is a measure of the relationship between two variables in the form of paired data items (*i.e.* $x-y$ co-ordinates on a graph). It is a real number in the range $-1 \ldots +1$. It is more usual to quote the *coefficient of determination*, $R^2$, necessarily ranging from 0 to 1, which indicates the proportion of the variation between the two variables that is explained by an underlying relationship. The rest of the variation is simply random noise.

However, it is possible, especially with small samples, that any apparent correlation arises simply through chance and that the real correlation between the variables is zero. For this reason, the significance level of the correlation is calculated. A correlation that is significant at the 10% level (termed $p < 0.1$) has a probability of less than 10% of arising purely through random chance.

The significance of the correlation is calculated using the *t-statistic*, which can also be used to calculate the significance of the Spearman's rank correlation coefficient ($R_s$). The probability given by the t-statistic indicates how likely the observed correlation coefficient would be if the true correlation were zero. The t-statistic depends on the number of degrees of freedom of the data, $n-2$, where $n$ is the numbers of paired data items. Two is subtracted to take account of the fact that more than

[1]Sometimes $R_{xy}$ to distinguish it from the Spearman Rank Correlation Coefficient

| | One tailed | Two tailed |
|---|---|---|
| $H_0$ | $r = 0$ | $r = 0$ |
| $H_1$ | $r < 0$ (or $r > 0$) | $r \neq 0$ |

Table B.1: One or two tailed significance test

two points are needed on a graph for any correlation to be derived: A straight line can always be drawn through two non-identical points on any graph. The t-statistic is calculated as follows:

$$t = r\sqrt{\frac{n-2}{1-r^2}} \tag{B.1}$$

where $r$ is either of the two correlation coefficients. The further away from zero the correlation coefficient, the higher the value of $t$ and the more likely any observed correlation is to be a true effect. Since $r$ can be $\pm 1$, it is possible for $t$ to be infinite. The significance test declares two hypotheses, $H_0$ and $H_1$, and can be either one-tailed or two-tailed, as shown by table B.1. The one tailed test is used if the direction of the relationship is known, *i.e.* if one of the variables is believed to be larger than (or smaller than) the other, rather than just being different from it. If nothing can be presumed about the nature of the relationship, then the two-tailed test is used.

This is converted into a probability value using a standard table of values published in statistics text books. These tables give the value that the t-statistic must exceed for any given number of degrees of freedom for the correlation to be genuine at a given significance level. The tables given are usually for two-tailed tests. If the test is a one-tailed test, then the same table can be used, but the column referred to in the table is for twice the significance level, *e.g.* if a one-tailed test is being carried out at the 5% significance level, the value required is read from the 10% (0.10) column.

The test of significance is carried out on the correlation coefficients derived from twelve trials of a genetic algorithm, giving ten degrees of freedom. Table B.2 shows the required values that the coefficient of determination must exceed in order to be significant at various levels.

| Significance level | Required t-statistic | Equivalent coefficient of determination |
|---|---|---|
| 20% | 1.372 | 0.159 |
| 10% | 1.812 | 0.247 |
| 5% | 2.228 | 0.331 |
| 2.5% | 2.634 | 0.410 |
| 1% | 3.169 | 0.501 |
| 0.1% | 4.587 | 0.677 |

Table B.2: Translation of coefficients of determination into significance levels for the genetic algorithm experiment.